2019

Pre-Travel Training And Real-Time Guidance System For People With Disabilities In Indoor Environments

Binru Cao

PRE-TRAVEL TRAINING AND REAL-TIME GUIDANCE SYSTEM FOR PEOPLE WITH DISABILITIES IN INDOOR ENVIRONMENTS

A Thesis Presented

by

BINRU CAO

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL AND COMPUTER ENGINEERING

May 2019

Electrical and Computer Engineering

PRE-TRAVEL TRAINING AND REAL-TIME GUIDANCE SYSTEM FOR PEOPLE WITH DISABILITIES IN INDOOR ENVIRONMENTS

A Thesis Presented

by

BINRU CAO

Approved as to style and content by:

____

Aura Ganz, Chair

____

C. Mani Krishna, Member

____

Russell Tessier, Member

____

Christopher V. Hollot, Department Head

Electrical & Computer Engineering

ACKNOWLEDGEMENT

I would first like to thank my advisor Prof.Ganz. This work would not have been possible without the help and advice of Prof.Ganz, who taught me a great deal about research and always give me suggestions whenever I ran into a trouble spot or had a question about my research.

I am highly indebted our lab member James for his guidance and constant supervision as well as for providing necessary information regarding this research.

I am also thankful to our lab member Zhuorui, Tao and Linlin. They provided very useful suggestions and help during my work.

ABSTRACT

PRE-TRAVEL TRAINING AND REAL-TIME GUIDANCE SYSTEM FOR PEOPLE WITH DISABILITIES IN INDOOR ENVIRONMENTS

MAY 2019

BINRU CAO

B.ENG., ZHENGZHOU UNIVERSITY, P.R. CHINA

M.S.E.C.E., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor Aura Ganz

Public transportation provides people with access to education, employment, health and community activities. However, navigating inside public hubs for people with disabilities such as cognitive or mobility impairments can be very challenging and dangerous. With the rapid development of digital technology such as Smartphones and sensors, there are unprecedented opportunities to assist people with disabilities to conquer these challenges.

In this research, we aim to create a two-step indoor navigation solution for users with different mobility and orientation abilities. In the first step, we developed a virtual reality-based pre-travel training module that enables users to familiarize themselves with the virtual environment which represents the physical environment. After users feel confident and familiar

enough with the environment, they proceed to the second step in which they visit the physical environment and use our real-time navigation assistance module.

The pre-travel training module is developed using a Unity-based 3D game and includes a virtual indoor environment that represents the physical environment. The game provides a navigation function that highlights the path between the user location and the chosen destination. Considering the unique needs of cognitive impaired users, we designed action training modules in the game environment which train the user to use the ticket machine, fare gate and call boxes. Such training modules help cognitive impaired users familiarize themselves with the environment as well as gain confidence to experience the physical environment.

When the users are ready to visit the physical environment, they use our real-time navigation assistance module which includes the same 3D virtual environment developed for the pre-travel training module. This approach is particularly important for people with cognitive impairment since they cannot to organize navigation cues effectively. Using the Bluetooth Low Energy (BLE) infrastructure in the environment, our localization algorithm can track the user location in real-time. Subsequently, the user's location will be integrated into the game environment so that the navigation path between the user's current location and the selected destination can be generated and visualized by the user on the fly.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The ability to independently navigate in urban environments is a fundamental necessity for all of us [1]. Outdoor navigation has been a well-established research area which is indispensable in enhancing people's sense of direction, particularly in an unfamiliar environment. Various GPS-based systems provide navigation instructions to guide users to travel from their homes to the public transit stations. However, these systems cannot help users after entering the stations since they neither provide internal navigation guidance to the desired subway platform nor do they provide extra assistance to board the right train, such as train arrival times. Therefore, we propose to develop an indoor navigation system which can be used by people to navigate inside the public transportation hubs. Currently, there are a few public transportation applications available to users, such as PVTrAck, which can provide real-time status about trains or buses to help users plan their trips. Based on the best of our knowledge, there is no available system aiming to navigate people to their destinations in a user-friendly representation (i.e., signs and alerts) with salient navigation cues along the path.

Moreover, with the growing number of senior and disabled people, the demand for a specially designed navigation system is increased dramatically. Nowadays, 8.5% of people (617 million) are aged 65 or over worldwide. According to the report [2], this percentage is projected to jump to nearly 17% of the world's population by 2050 (1.6 billion). On the other hand, more than one billion (15%) people in the world live with at least one form of disability, of whom 2-4% experience significant difficulties in functioning [3]. Most of these disabilities have adverse effects on the ability to navigate and orient. Almost all of the current wayfinding and navigation systems are designed for people without disabilities by concisely providing navigation instructions. However, for aging and disabled people, such navigation aids are not sufficient since they do not have good spatial cognition and navigation skill. The reasons that cause these issues can be various. Some of these people are suffering from short-term memory lapses, lacking planning ability [4] and limited communication ability. Others are suffering from a mobility problem.

No matter what their conditions are, seniors and people with disabilities have issues with navigation assistance, especially in the indoor environment. Depending on their requirements, navigation assistance for these people need to focus on usability, intuitive interaction, and natural learning. The instruction must be simple and straightforward by providing a precise and reliable way to deliver information to the users.

Based on these requirements we build a system which provides indoor navigation assistance for seniors and disabled individuals with two modules. Our system includes a pre-travel training module using a virtual environment, and then a real-time guidance module in the physical environment using real-time localization with the virtual game as the background.

In chapter 4, we introduce the pre-travel training module that will help people with disabilities familiarize with the environment before they arrive there physically. The pre-travel training module is built using Unity3D engine and presents the physical environment to our users through the game environment. Our virtual game contains some unique features for people with disabilities. First, we used highlighted path and indication arrows to inform the cognitive impaired users the way they should follow to reach their selected destination. Second, we introduced the action training function by which cognitive impaired users can practice how to operate the facilities in the space, such as the ticket machines or the fare machines.

In chapter 5, we covered our real-time guidance module. First, we migrated the video game from PC end to the mobile end, and then users could use this mobile game when they are in the physical space. To leverage the established familiarity between users and the game environment, we overlaid the real-time locations of the user onto the game environment. User's location is determined by our localization algorithm in real-time using BLE technology. By having real-time location information of the user, our guidance module can assist users while they are traversing in the physical environment. We believed that the knowledge about environment learned from pre-travel training module can significantly help our users understand the real-time navigation assistance in the unfamiliar environment.

CHAPTER 2

RELATED WORKS

There are many existing navigation systems for different disabilities. We categorized disabilities roughly into two main classes. One is mentally impaired; another is physically impaired.

Cognitive impairment, the most important type of mentally impaired, ranging from ones that are present at birth (such as Down's syndrome, mental retardation and cerebral palsy), to ones that are acquired due to some form of traumatic brain injury or illness (such as aphasia, a speech and language disorder, or amnesia), to ones that emerge through the normal aging process (such as Alzheimer's disease), to ones that arise due to complicated causes such as schizophrenia [5]. Cognitively impaired are very sensitive to abstract information and may not be able to recall clues of routes when they were traveled. Many researchers presented assistance systems for this group to help them travel either outdoors or indoors safely and independently.

Table2.1: Compare box for different assistance navigation systems with ours

Criteria	[6]	[7]	[8]	Our system
Real-time/Pre-travel	Real-time	Real-time	Real-time	Both
Platform/Device	PDA	PDA	Mobile device	Laptop & Mobile device
Landmarks	RFID tags	QR-code	QR-code	BLE beacons
Navigation Instruction Delivery Method	Photo with directions	Photo with directions	Photographs showing the position and view of next QR-code	3D training game
Localization/Tracking technology	RFID readings	Tracking by PDA device scanning the QR-code	Scan the nearest QR-code	BLE indoor localization
Environment map	Photographs	Photographs	Blueprint	3D Virtual environment

Paper [6] gave a solution for indoor environment, which used RFID tags as landmarks. Instruction delivered as a photo with direction information to users when they were in the range of RFID tags. Users tracked by position information from RFID readings. Both [7] and [8] used QR-codes as landmarks. In paper [7], the user got the navigation instruction or position information by scanning the nearest QR-code - navigation instruction delivered as a photo with direction information when the user scan a QR-code. Navigation instruction delivery method in the paper [8] is different from the paper [7]. In paper [8], the navigation system delivered as a photograph showing the position and a view of next QR-code.

Mobility impairment is the dominating type of physical disability, which a wide range of world population suffered from. Such disabled people found it challenging to navigate themselves even in the indoor environment, which made them isolated and in pursuit of external assistance. Lots of research are trying to find an efficient way to help these disabled people navigate themselves independently in daily life. Paper [9] described a navigation system with a framework based on the creation of a road graph with different levels of details (LoD). Sidewalks, crosswalks and dynamic obstacles have been considered during the planning to enable safe planning for physically impaired people. The obstacle detection module in this system helped to detect the obstacles and change the current path for users. Paper [10] described a navigation system using voice assistive technology to aid physically disabled people to access commonly used places in their homes or any other particular room. With a voice capture and recognition module, the system can recognize speeches and give the corresponding instructions. The system in the paper [11] designed an “Access Feature” to define a step-free or accessible zone for wheelchair users.

CHAPTER 3

SYSTEM ARCHITECTURE

In this chapter, we will introduce our system from a general picture.

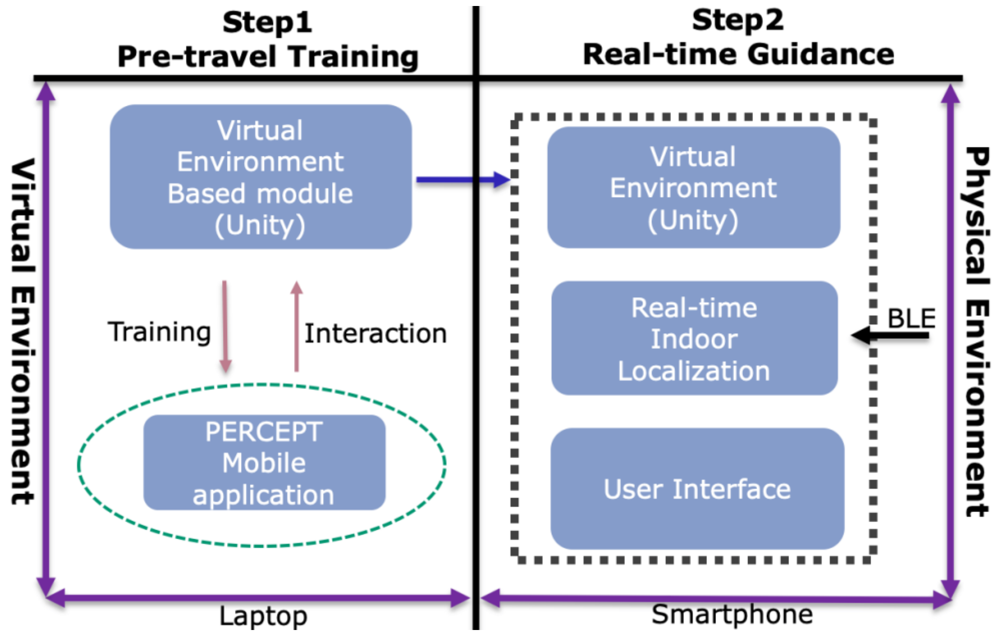

Figure 3.1: System Architecture

We created a two-step indoor navigation solution for seniors and users with different mobility and orientation abilities. Step1 is a virtual reality-based pre-travel training module, which enables the users to familiarize with the virtual environment that represents the physical environment. Step 2 is a real-time guidance assistance module, which provides real-time guidance assistance in the physical environment after users feel confident and familiar enough with the virtual environment.

Pre-travel training module included a virtual environment-based game using Unity3D platform which represented the environment of Boston North Station. The game provided navigation services that user can do practice in traveling from one place to another place. The user can explore the whole environment as many times as they want. Users will be acquainted with the environment before they get there.

We also provided interaction between the game and PERCEPT mobile application. The user can only use PERCEPT application in physical space because PERCEPT must work with deployed BLE tags as landmarks. Pre-travel training game has the virtual landmarks which do not need the real BLE tags being deployed. It can provide pre-travel training for the user to learn how to use the PERCEPT on the smartphone proficiently and help them to build the mental map themselves.

After the user gets familiar with the virtual environment, they are ready for the physical space. In order to keep assisting the user in a physical environment, we built a real-time guidance module. The real-time guidance module included three pieces: a. The virtual environment, which leveraged the work in step 1. b. Real-time indoor localization using BLE technology, which provided a real-time localization function to track users' position during moving. c. A user interface, which allow the user to select the desired destination and get navigation instructions.

CHAPTER 4

PRE-TRAVEL TRAINING USING VIRTUAL ENVIRONMENT

Due to population aging, the mentally impaired rate increases extremely fast. Assistive navigation tools for this group are in urgent need. Compared to ordinary people, seniors and people with cognitive impaired have difficulty to understand the environment because of their bad spatial and recognition skills and short-term memory lapses. The most critical assistance is helping them to establish a mental map of the environment.

4.1 Virtual Environment Based module

4.1.1System Architecture

The virtual environment based module has several components:

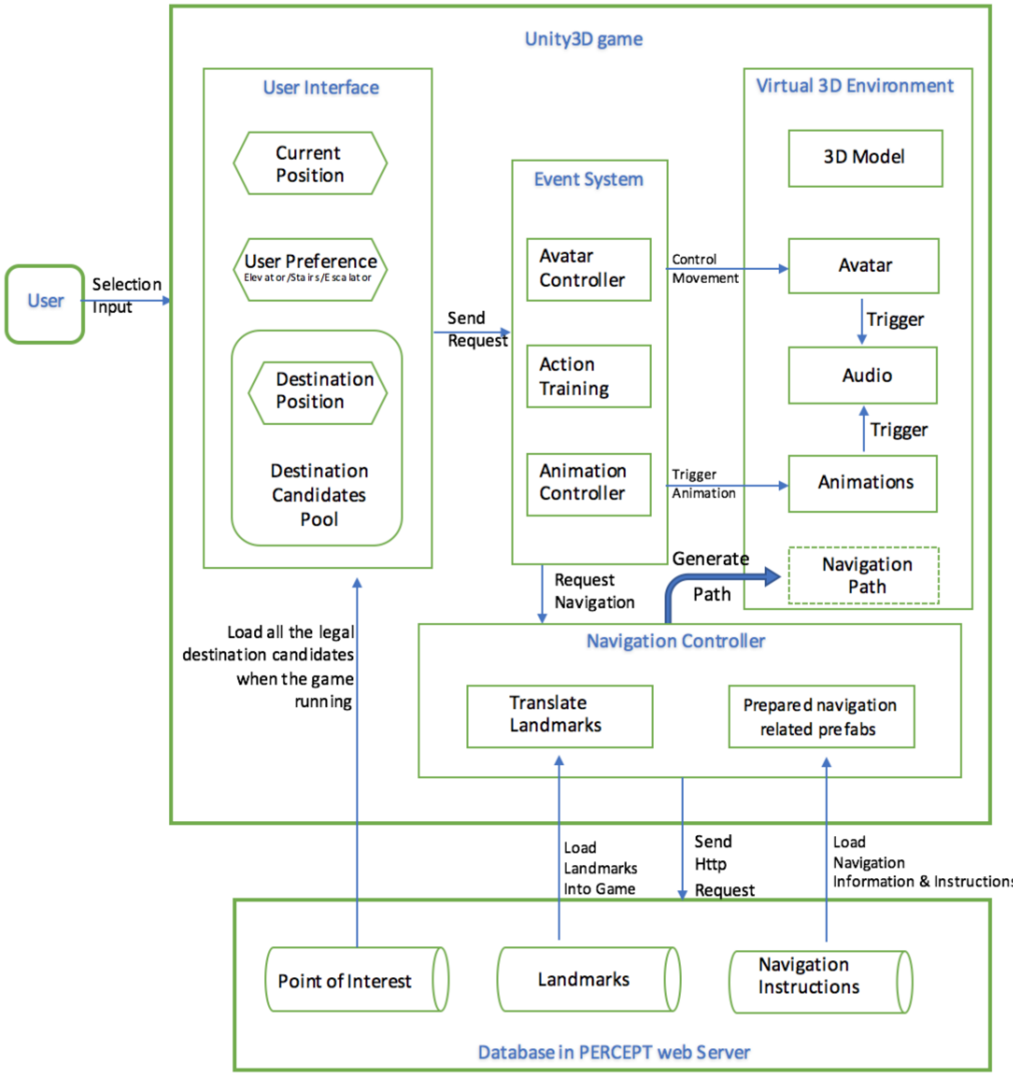

Figure 4.1: Pre-Travel Training System Architecture.

Virtual 3D Environment:

- 3D Model: The virtual 3D environment model was built by the blueprint of Boston North Station.
- Avatar: The game character controlled by the user when they are playing the game.
- Audio: We supported different types of system sound in the game such as indicator sound and footstep sound to help the user get more spatial information from surroundings.

- Animations: We created animations like “Door open”, “Door close”, “Escalator up”, and “Escalator down” to simulate the physical-world scenario.
- Navigation Path: After user selected their destination and preference, the event system would trigger the navigation controller to send an HTTP request to PERCEPT web database to retrieve path information, then a visualized path will be generated and overlaid on the game environment.

User Input & Interface: Users can select three terms.

- Start location: Included two difference main entrances of North Station.
- Destination: Included serval valid locations in the station.
- User Preferences: Included Elevator, Escalator, Stairs.

Navigation Controller:

- Translate Landmarks: The information about landmarks that retrieved from PERCEPT web database is stored as physical world coordinates, the navigation controller needs to translate it to unity coordinates for the system to understand.
- Navigation Information and Instructions: All the navigation instruction information that we used is stored in the database on the PERCEPT web server. Based on the selected start location and destination with user preference, the system would send an HTTP request to the server and retrieving the related navigation information to the game.
- Prepared prefabs: Prefab is a useful asset type in Unity; it acts as a template and can store any game object complete with components and properties that we set in the previous. In order to assist our aimed user group to follow the instructions while moving smoothly, we set each part of the path to be highlighted with red arrows. This intuitive and straightforward way can effectively decrease the user’s obfuscation. We created two

prefabs for generating a highlighted path. One is the landmark; another is the link between these landmarks.

Event system: All the game objects' action events are registered through the event system. Whenever the trigger attached to the object has been touched off, this object will give the corresponding response. There are three types of event in our system:

- Animations controller: We used several different animations attached to game objects to enhance the user immersion experience.
- Action training in functional areas: We defined some functional areas in our game sense. When the user step into any of these functional areas, the event system will jump to a corresponding game scene.
- Avatar controller: User could use the touchpad to control the avatar's movements in the game.

The database in PERCEPT web server: All the data we need is stored on PERCEPT web server's database. We can download the data by sending an HTTP request to matching URL to query them.

4.1.2 Implementation Details

- Development platform and tool

Our pre-travel training virtual game is developed on the Unity3D platform, which is a powerful and versatile tool to achieve interactive experience for users. It is an all-purpose cross-platform game engine developed by Unity Technologies, which primarily used to develop games and simulations for computers, web player and mobile devices [12]. It offered plenty of useful built-in components like Game Object, Materials, Texture, Animation, Models, Graphics rendering, Lighting, Physics collider, Script and user-friendly interface for development. Unity3D supports drag and drop functionality and scripting through C# and JavaScript. Many other .NET languages can also be used with Unity if they can compile a compatible DLL. The script is an essential ingredient part of all games; its work is responded to input from the player and arranged for events in the gameplay to happen when they should. Beyond that, scripts can be used to create graphical effects, control the physical behavior of objects or implement a custom AI system for characters in the game [13].

- Environment Creating

Our virtual environment represents the exact physical environment of the target building, which is Boston North Station. There are several steps to create a virtual environment: Generate a 3D model in Google Sketch Up according to the building's blueprint and import it into Unity game sense; Attach physics components like Rigidbody and Collider to some necessary game objects in order to make them move realistically; Add animations. The details of the last two steps list below.

I. Attach physics components.

The rigidbody component enables the game objects to act under control of physics. The Rigidbody can receive forces and torque to make objects move in a realistic way. By attaching the Rigidbody to the avatar, it will act like a human being, which is walking under gravity and only spin around y-axis (i.e., the vertical axis in the space) and keep the other two axes untouched.

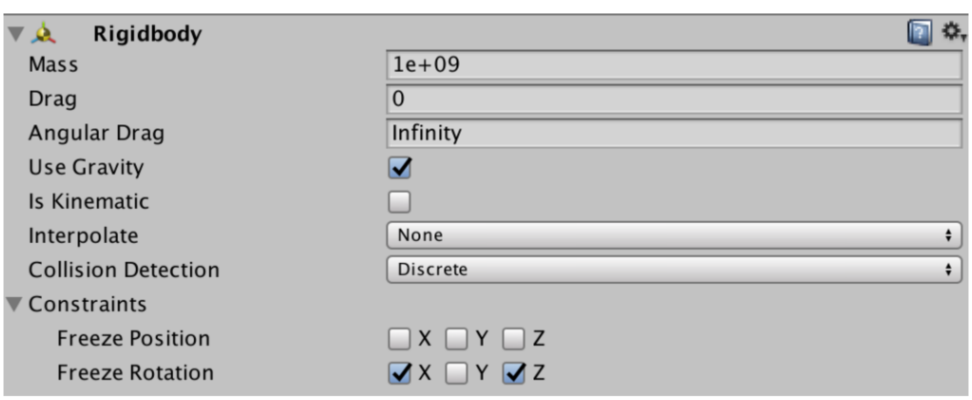

Figure 4.2: Rigidbody component in Unity.

The collider is another important physic component. It defines the shape of an object for the purposes of physical collisions. There are many types of collider such as Box Collider, Sphere Collider, Capsule Collider and Mesh Collider. In our game, every item in the 3D model has attached a collider component with “is Trigger” unchecked. In that way, those objects will not be able to pass through. We also use empty collider objects to set some conditional virtual gates.

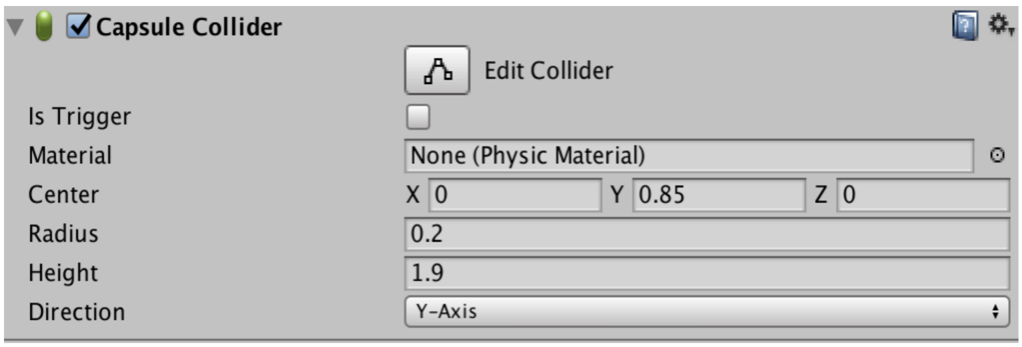

Figure 4.3: Collider component in Unity (capsule collider as example).

II. Add animations

We used two different ways to create animations inside our game. One is using the Unity scripting system; another is using Unity animation system.

Scripting is an essential ingredient in Unity. All game need scripts to respond to the input from the player and handle the events in the gameplay. Compare to the animation system, using the script is more flexible for the developer to create and develop a simpler-level animation like we do the “Fate gate open” and “Fare gate close” animations. There are four steps to accomplish this.

Step1: Create a new script and initial all the variables

Step2: Using “OnTriggerEnter” and “OnTriggerExit” interaction functions of collider to change the position of two sides of the gate continuously.

Step3: Update the position of each gate in every frame.

Step4: Attach this script to an empty box collider, and set the left gate and right gate component separately with the corresponding objects in inspector.

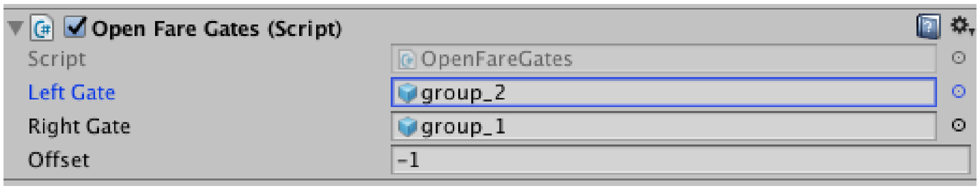

Figure 4.4: Script component for creating the animation

For creating the complex animations, it is better to use the Unity animation system since it provides easy workflow and setup of animations for all elements of Unity. It allows animators to work more independently, so developers can preview the animations before the gameplay code is hooked in and manages complex interactions between animations with a visual programming tool. The Unity animation system is based on Animation Clips, which contains information about how objects should change their position, rotation or other properties over time. We take the “Elevator door” animation in our game as an example to explain the whole process.

Firstly, we generate 6 animation clips called “BottomDoor1Close”, “BottomDoor1Idel”, “BottomDoor1Open”, “UpDoor1Close”, “UpDoor1Idel”, “UpDoor1Open”. Each clip defines an object’s changing properties.

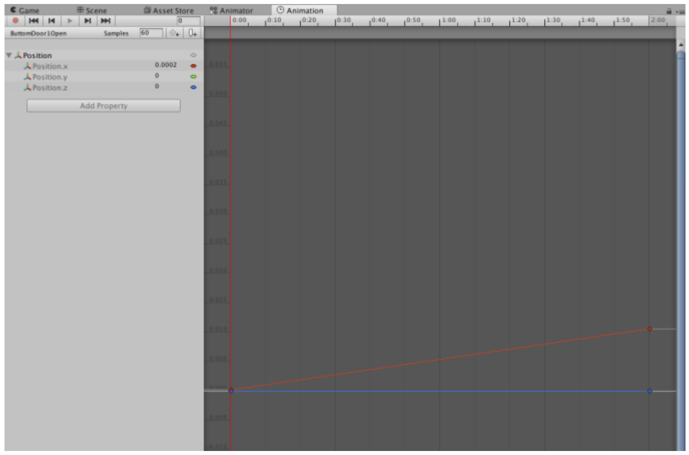

Figure 4.5: Animation clip window

Next, all these animation clips will be integrated together sequentially into an animation controller. Animation controller acts as the “Status Machine”, it keeps track of which clips should be playing or blend together. In the “elevatorUpDoor” controller, there is total 3 clips (shown below), and has two parameters “doorOpen” and “doorIdle” as the conditions which will affect the transitions between clips and determines which should play in the next play frame.

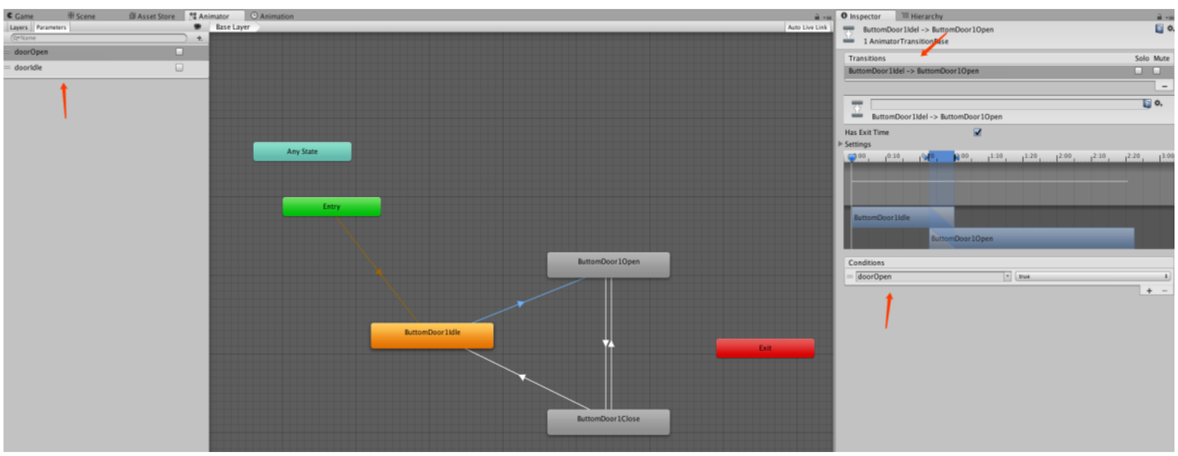

Figure 4.6: Animation controller window

- Action Training

Our game extended a unique functional feature called "Action Training" to help disabilities to familiarize themselves with the environment. There are many actions could be happened when people in a subway station, such as using a ticket machine to purchase a ticket, swiping the ticket to pass through a fare gate, using an emergency call box for help and so on. We provide the simulation of these possible actions in our game using the game canvas system.

Here we use the example "Purchase ticket" to explain the details. How to purchase a ticket may be the first challenge for seniors and cognitively impaired people in a subway station, especially a station they never been there before. Our game let the user experience and practice the whole purchase process. The process is showing below.

1. At the beginning of the game, the user's ticket balance shows "$0.00. We set a transparent collider just in the front of fare gates, and if the user wants to go through the gate, it should satisfy the conditions that can trigger the collider to open. When the ticket value less than $0, the collider will not be able to go through, and the player will be stopped outside. Only when the balance is greater than $0, the collider will be triggered to be opened and let the player pass through.

Figure 4.7: Virtual Ticket with balance.

2. The user needs to buy a ticket or add value to an exits ticket in order to pass through the fare gate. We created a button called “back to ticket machine”, the user can click this button to move to a ticket machine immediately, or they could control the character to walking back to a ticket machine. Each ticket machine has its functional area (the green box) called “Ticket Machine”, when our avatar steps into this area and facing to the ticket machine, our purchasing interface will open automatically.

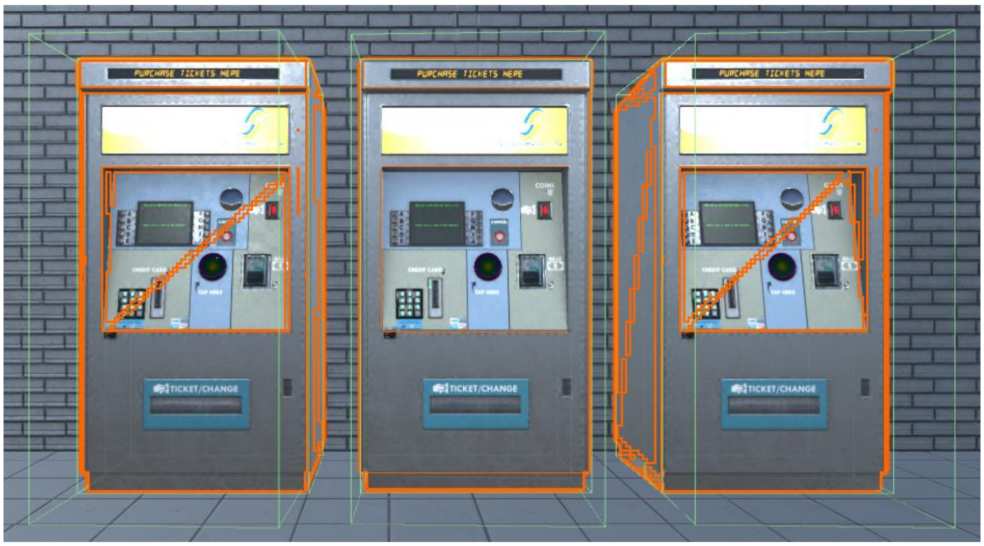

Figure 4.8: Functional area of ticket machine.

3. This is the welcome page of purchase scene. We create this interface precisely the same as the Boston Subway Station purchasing system. Tap “Press Here” to continue. Tap “Exit” to exit this purchasing system.

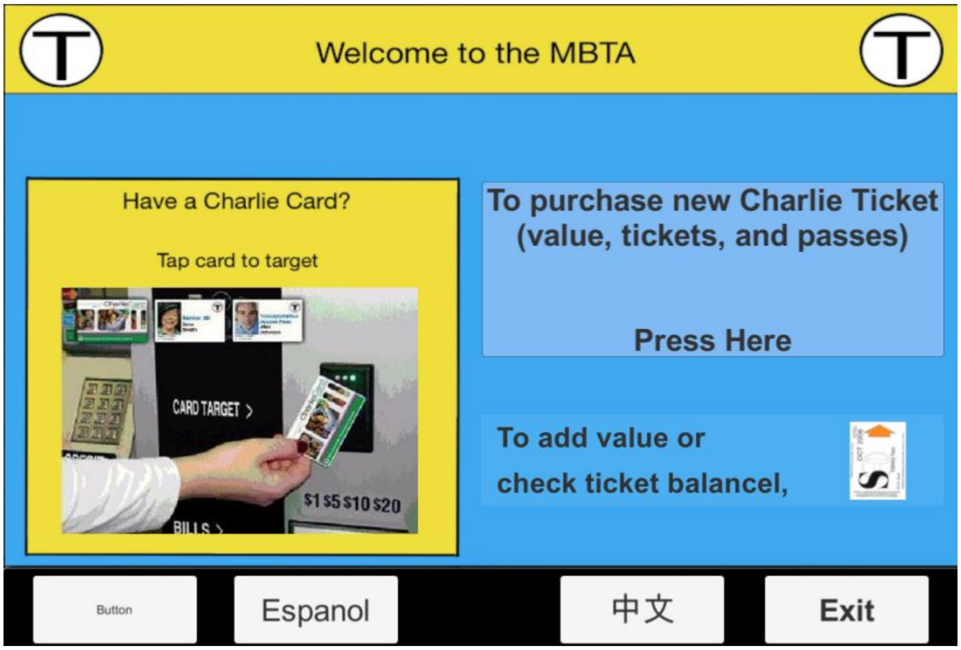

Figure 4.9: Purchasing interface.

4. Tap the “Bus & Subway Tickets” button to continue, tap the “Go Back” button if the user wants to change their previous selection, tap the “Cancel” button if the user wants to cancel this purchase and return to the welcome page.

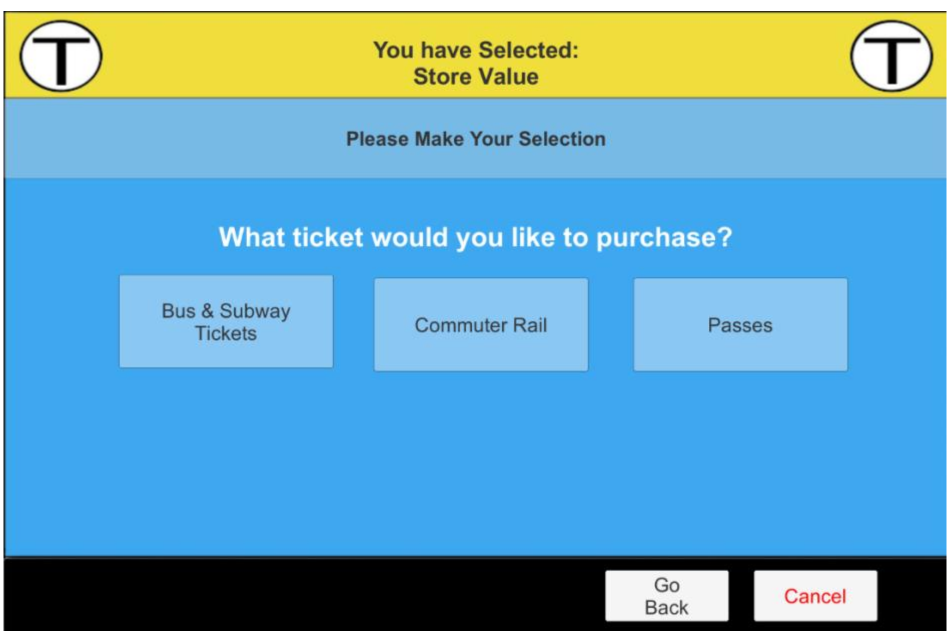

Figure 4.10: Purchasing interface.

5. There is a total of four types of fare, the senior and disabilities using the same “Senior CharlieCard”. Tap “Senior” or “Person with Disabilities” to continue. Tap the “Go Back” button to change their previous selection, tap the “Cancel” button to return to the welcome page.

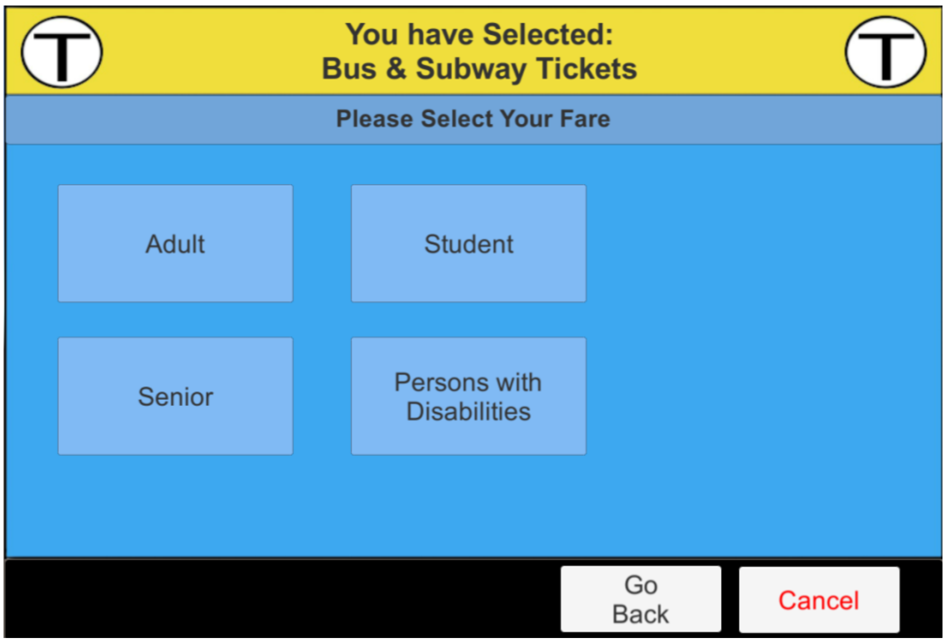

Figure 4.11: Purchasing interface.

6. Tap the “One Way” button to add \$1.10 value to the virtual E-ticket, tap the “Monthly” button to add \$30.00 value to the virtual E-ticket.

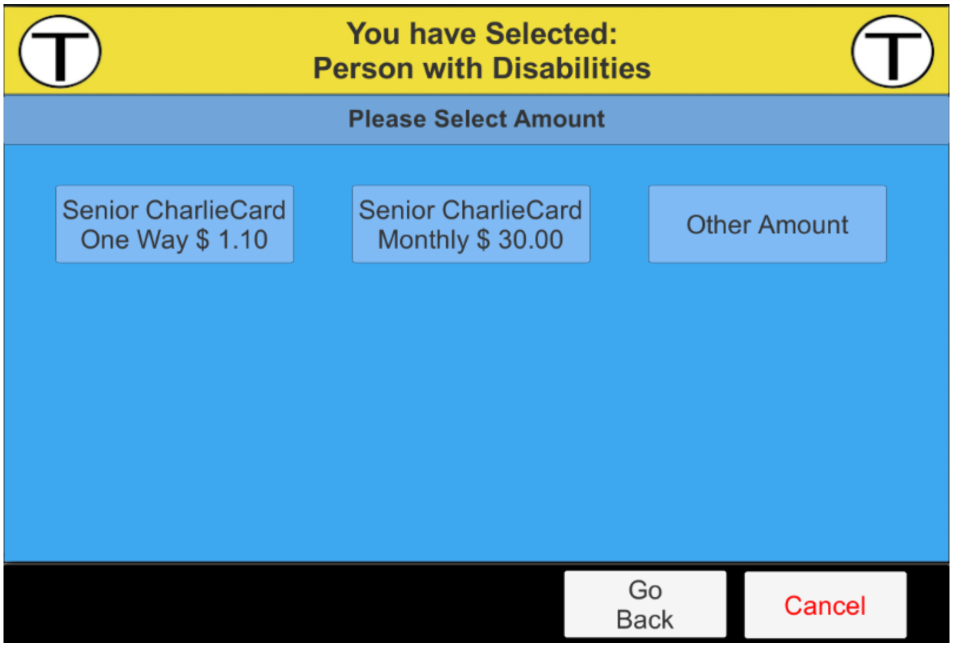

Figure 4.12: Purchasing interface.

7. As an example, we tap the "One Way" button and store the value to the ticket. Tap the "Confirm" to finish and jump back to welcome page. Tap the "Go Back" button to change their previous selection, tap the "Cancel" button to return to the welcome page without neither purchasing ticket nor store value.

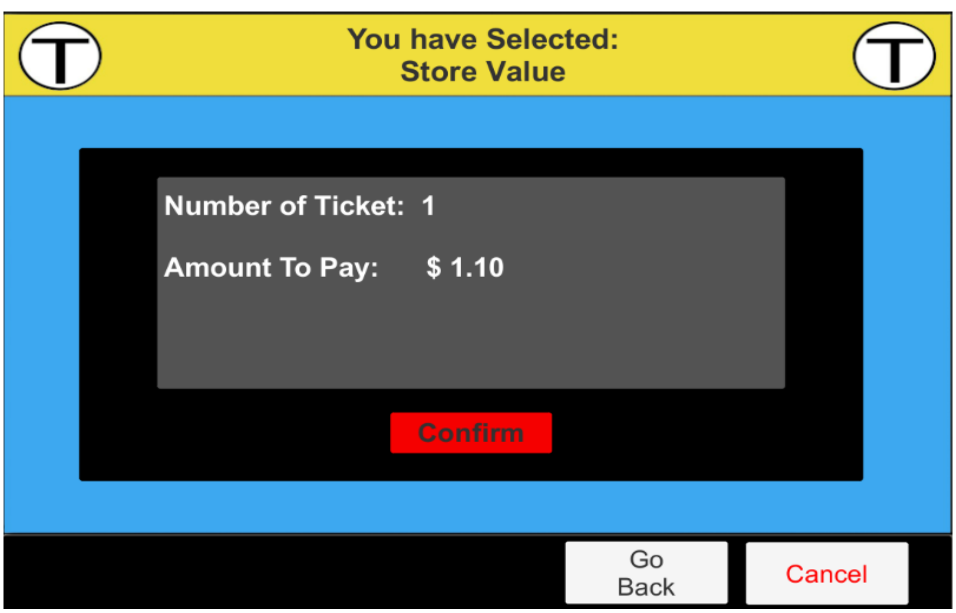

Figure 4.13: Purchasing interface.

8. After finish buying the ticket, our ticket balance will be changed depending on the value that the user purchased. Then when our avatar comes again to the fare gate, the collider is triggered as open.

Figure 4.14: Check the virtual ticket with balance again.

- Navigation using the visualized path

Before the user starts to play the game, there are many options for them to select to satisfy their requirements. We supported instruments option includes Elevator, Stairs, Escalator, and destination option includes some valid destination candidates. Considered the senior, physically impaired and wheelchair user might not be able to use the regular route that involved stairs, we provide elevator and escalator option to make their travel more convenient. Our server has stored many possible valid destinations when the game starts running. It will download all these destinations automatically for the user to select. Depend on user's choice of the instrument and destination, the game event system will get a prompt and send an HTTP request to our online server to search and retrieve the correct information.

According to papers, the seniors and cognitively impaired people suffered some difficulties in navigation process and carried out tasks. First, they have terrible spatial and orientation skills, when in the navigation process, lacking these skills will lead the user to make a turn at a wrong place and incorrect orientation. Another problem they usually have is the short-term memory lapses, bad recognition skill and oblivion to essential tasks, a long and complicated instruction will have less help for them. The requirements for these users are quite clear and definite. Firstly, they need good intuitive interaction. Secondly, the navigation information should be reliable, safe, simple without abbreviation and easy to learn. Thirdly, infer the required information automatically. In consideration of all the problems list above, we created some visualized indications (i.e., images) to show the whole path straightforwardly. These two kinds of indications that we used in the game are shown below:

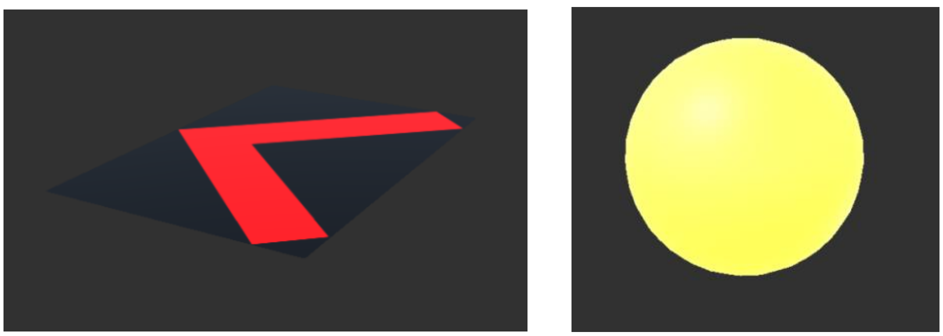

Figure 4.15: Prefabs for path. Left: red arrow represents the links. Right: yellow sphere represents the landmarks

The yellow sphere represents the necessary landmarks during the path. The landmark's information came from the data of navigation instructions that we loaded from the online server. A list of red arrows uniformly distributes between two adjacent landmarks, the arrow directed from the previous node to the next one, indicated the path in a visualized way. The user could follow the list of arrows to arrive at their destination correctly and efficiently without worrying about losing in the path. Our visualized instructions will not bother user to determine where to go in a decision point comparing with audio or contextual instruction, so they do not need to take too much head work, which makes their life much more comfortable.

All the path and landmark information we received is stored as longitude-latitude coordinates. In order to represent the path correctly inside Unity game, we must translate coordinate between the geo-coordinate system and Unity coordinate system. In Unity, y-axis holds for height, so when translating coordinates, we only consider x-axis and z-axis as x and y in Cartesian coordinates. There are two steps to translate coordinates.

1. Translate and scale the coordinates.

   We use a pair of coordinates for the same point as the datum point:

   (1482.4, 191) ⇔ (42.3658521162533, -71.0608604550362)

   At the equator, a degree of latitude is 69 miles (1113000m), and a degree of longitude is cos (latitude in decimal degrees) * length of degree(miles). So, we translate the position based on that translation rule.

$$Distance\_x = latitude * 111300 \tag{4.1}$$

$$Distance\_z = longitude * 111300 * \cos(latitude) \tag{4.2}$$

2. Rotate the translated the coordinates.

   There is a rotation matrix for this kind of translation:

$$R = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \tag{4.3}$$

   The θ comes from the rotation angle from one coordinates to another. In our case, the angle is 125 degrees.

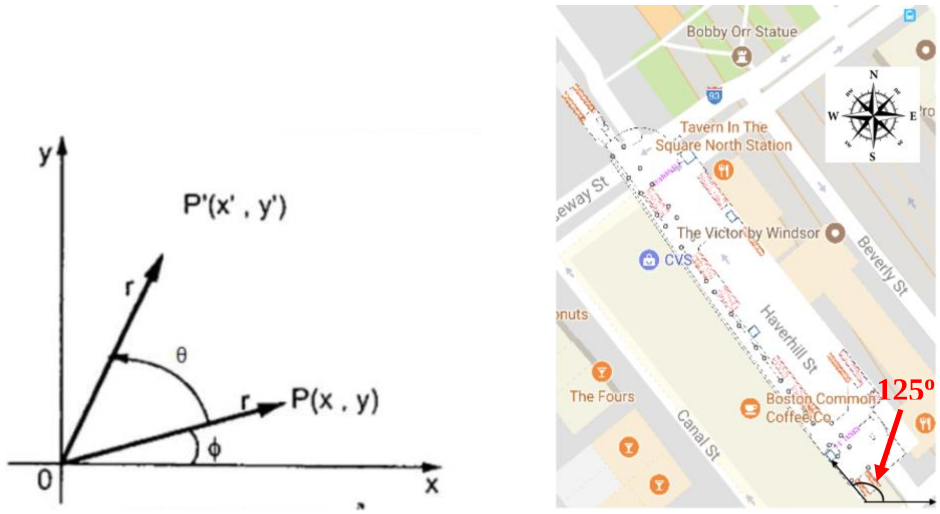

Figure 4.16: The rotation angle used in our system.

After we translated all the landmarks into the game, we will get a visualized path inside the game showing in Figure 4.17.

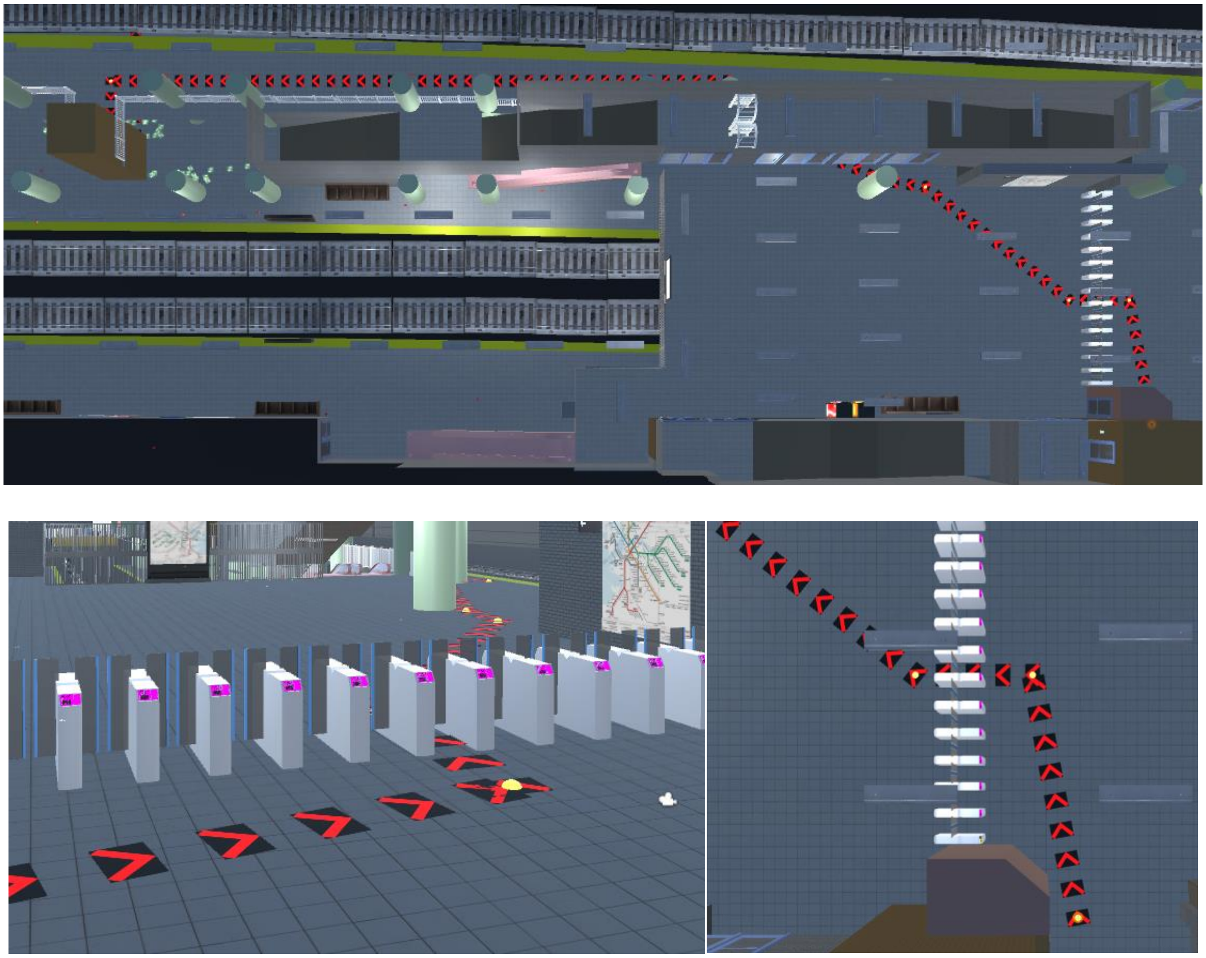

Figure 4.17: The visualized path inside game.

- Textual Navigation Instruction

In addition to the visualized path, we provide textual instruction as the supplementary assistant. As mentioned before our navigation instruction path is generated by a chain of landmarks. The textual instructions came with each landmark. Every landmark in a specific path comes with a corresponding instruction sentence. When the system generates the landmarks in the visualized path, the instruction sentence for each landmark can also be retrieved at the same time. We set a trigger sphere for each landmark, coming with two trigger functions called "OnTriggerEnter ()" and "OnTriggerExit ()". When the user came into the trigger range of the landmark, the function "OnTriggerEnter ()" will be called automatically so that the instruction sentence will pop up in the upper left corner of screen. Once the user steps out this landmark's trigger sphere, the "OnTriggerExit ()" function will be called, and the instruction sentence will disappear.

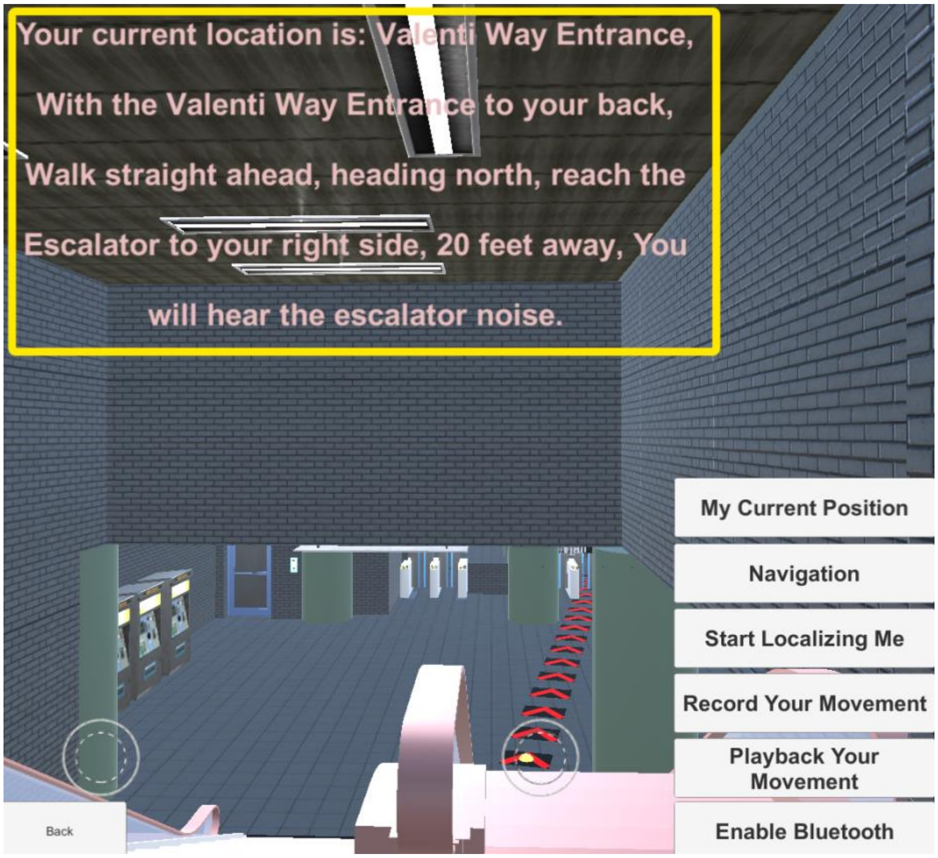

Figure 4.18. The textual Navigation Instruction.

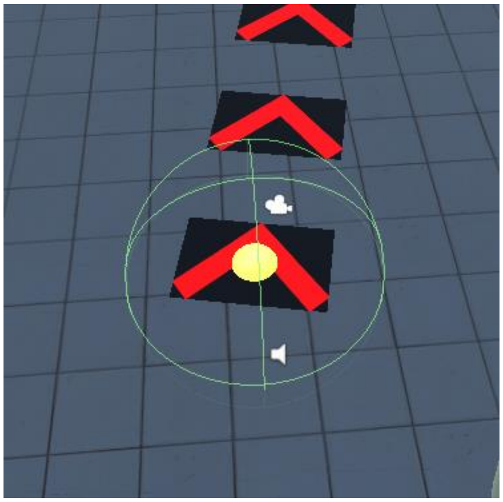

Figure 4.19: Trigger sphere of landmark.

- Record and Playback function

Another useful function in our game is called “Record and Playback service”. This function works both on the training session and real-time navigation session. It is beneficial when the user or their trainer wants to playback their entire action process when using the navigation instruction system. This function can provide valuable data for user behavior analysis for their training and practicing reason. Be specific to our aimed user group (mobility or cognitive impaired) this function can help them find out the reason of failures and improve their ability to understand and follow the instruction, which will significantly increase the success rate when our user use our navigation service in real-time.

We split this function into two phases. One is the “Record” function; another is the “Playback” function. The “Record” function will run once the game start to launch. It will track the avatar’s every movement including both position and rotation data. When the user finished their route or even lost during the navigation, they can save it for later review by just clicking the “Record Your Movement” button. All the movement information so far will be recorded in the backend. Unity has a special tool called “PlayerPrefs” allows saving and loading game data without external files. We write the route data into “PlayerPrefs” to save it internally. When clicking the “Playback Your Movement” button, the game engine will load the saved route in “PlayerPrefs”. The avatar will move according to the data from the record route and show the entire process of the last saved movement.

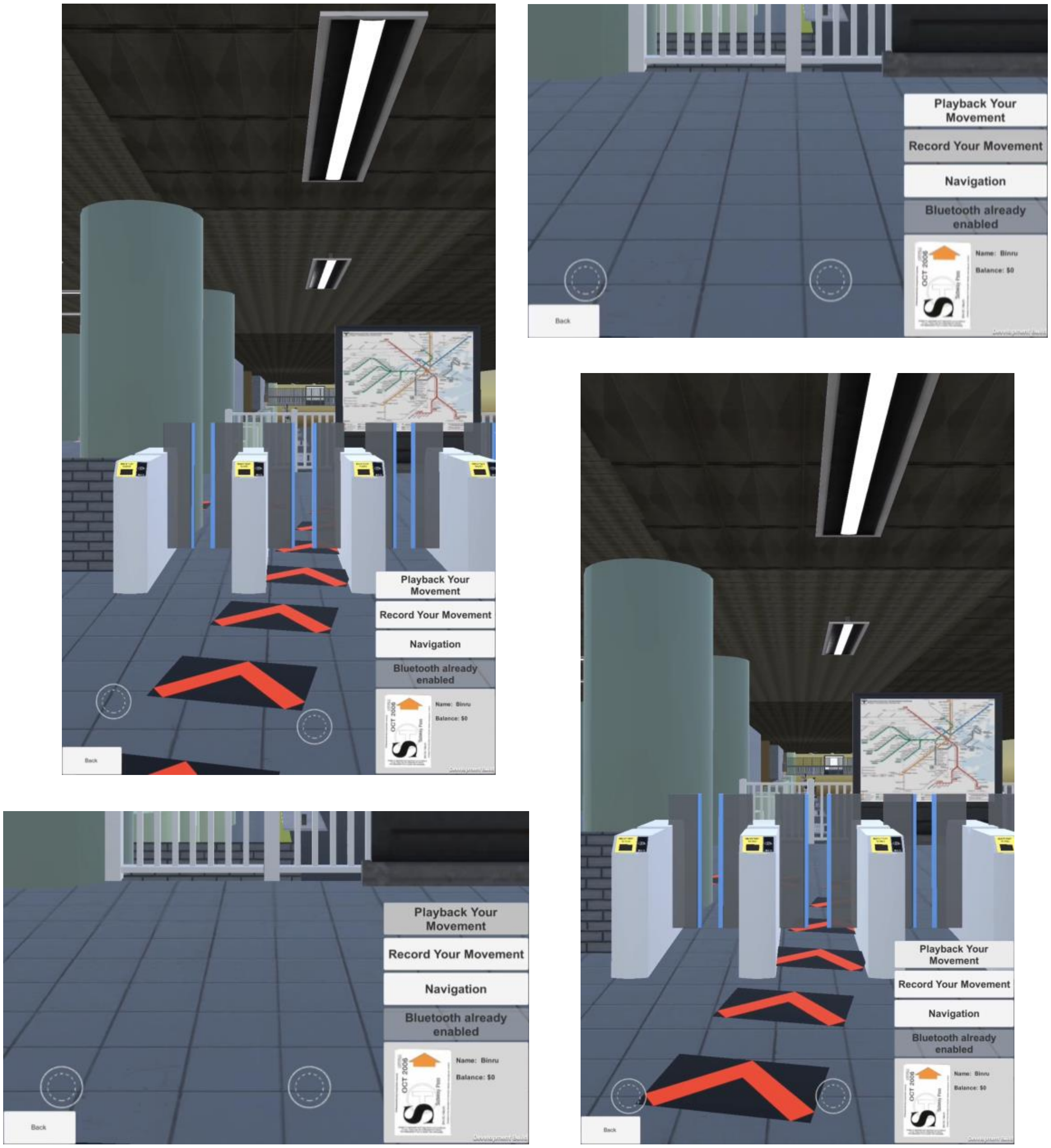

Figure 4.20: Record and Playback service

Left top: Record mode, control the avatar by joystick.

Right top: Click the “Record” button to save the movements.

Left bottom: Click the “Playback” button.

Right bottom: The avatar playback the saved path automatically.

4.2 Interaction with PERCEPT

4.2.1 Motivation and Work flow

Another critical purpose behind our pre-travel game is training user to get familiar and proficient use PERCEPT indoor navigation application on the smartphone before they go the physical environment. PERCEPT only can be used in real-time, working with deployed BLE tags in the environment as landmarks. Our 3D virtual environment has virtual landmarks which do not need physic BLE tags being deployed. It can provide pre-travel training for the user to practice how to use the PERCEPT on the smartphone proficiently. Also, it gives the user a more intuitively understanding of environment and surroundings, which helps the user build a mental map.

The workflow of the interaction between PERCEPT and the pre-travel game is in Figure 4.22.

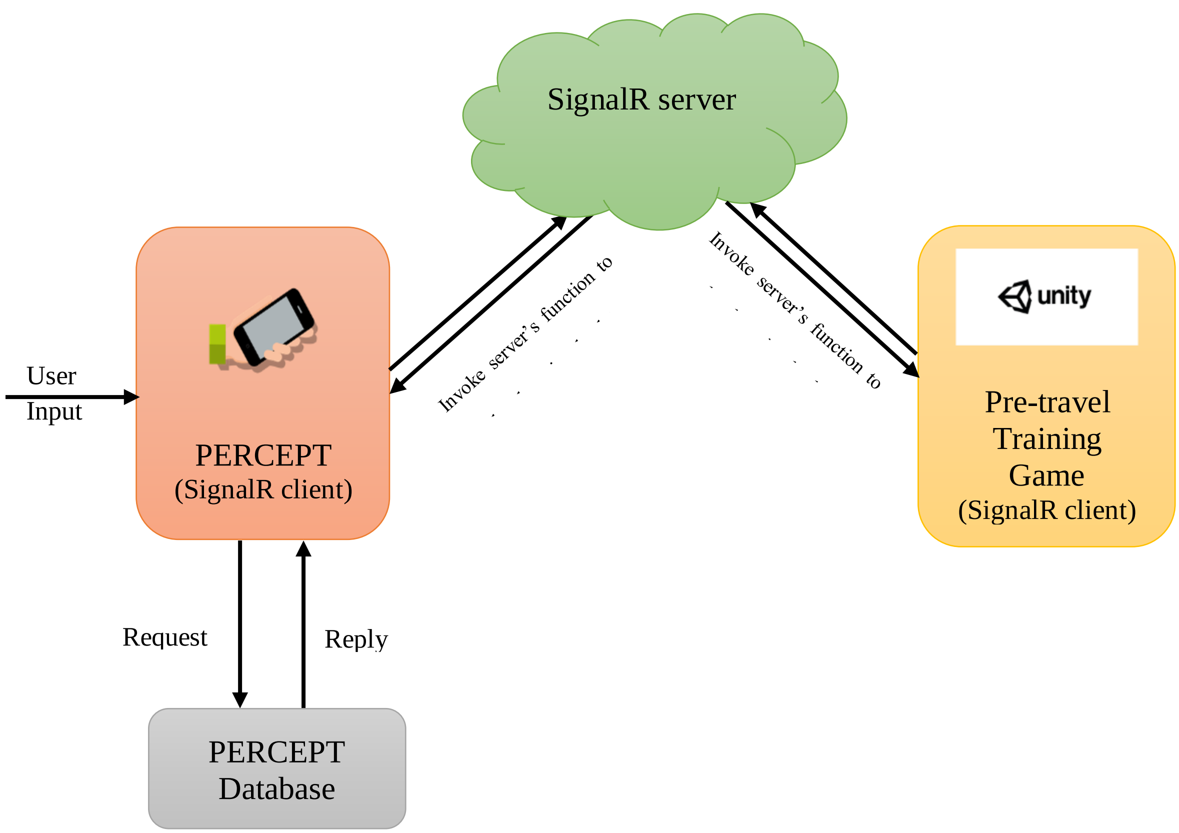

Figure 4.21: Work flow of interaction between PERCEPT and pre-travel training game.

4.2.2 Implement Details

ASP.NET SignalR is the tool we used to build the connection between PERCEPT and pre-travel training game. SignalR is an ASP.NET library that allows bi-directional communication between server and client. It can develop real-time web functionality, which is the ability to have server code push content to connected clients instantly as it becomes available rather than having the server wait for the request from clients. The concept of connections between server and clients is in Figure 4.23.

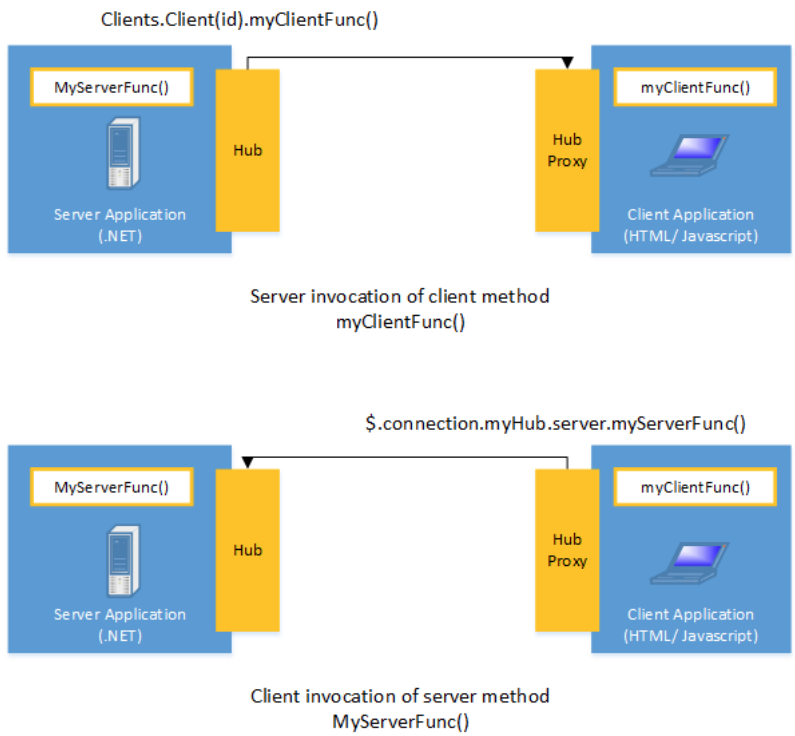

Figure 4.22: The invocations between server and client.

We built a SignalR server (http://signalrtoazure20170617053409.azurewebsites.net/) on Windows because it only can be hosted on Windows operating system. We named our server hub as "MyHub", and created a server function called "Send". "Send" method declared public methods on a hub so that clients can call them. The web console in Figure 4.24 can record all the data exchange between server and clients. Figure 4.25 shows information about our SignalR online server. This window shows the setting information we used for SignalR communication. On the top of the picture shows the necessary user setting information includes name, status, location, subscription ID and the URL of this server. The “Data In” window indicates the received data from clients, and the “Data Out” window indicates the sent data from this server to other clients.

Figure 4.23: Web console.

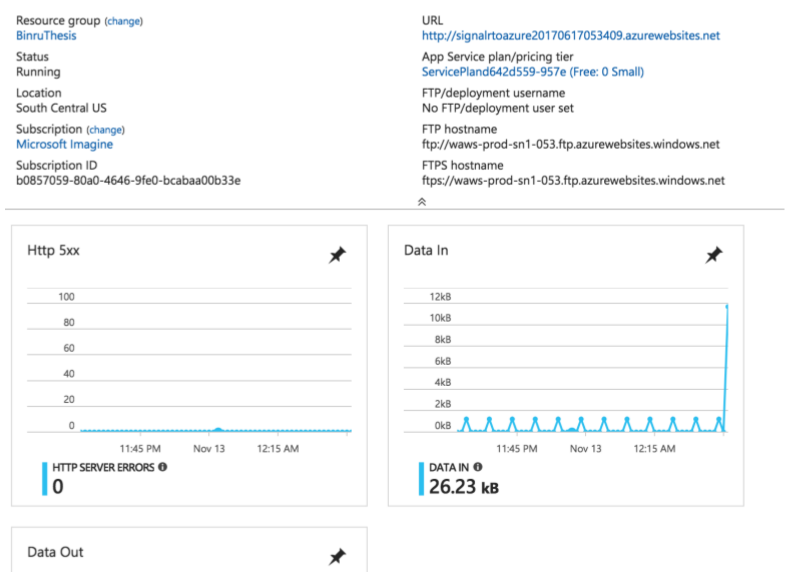

Figure 4.24: Data about the server.

Both PERCEPT and pre-travel game are clients. Each client created a “HubConnection” object with the site URL we mentioned before and created a proxy to listen to the callback function from the server after connected to it. To establish the connection client called start function of

“HubConnection” and wait for connecting. Pre-travel game and PERCEPT can make real-time communication after both of them connecting through server successfully.

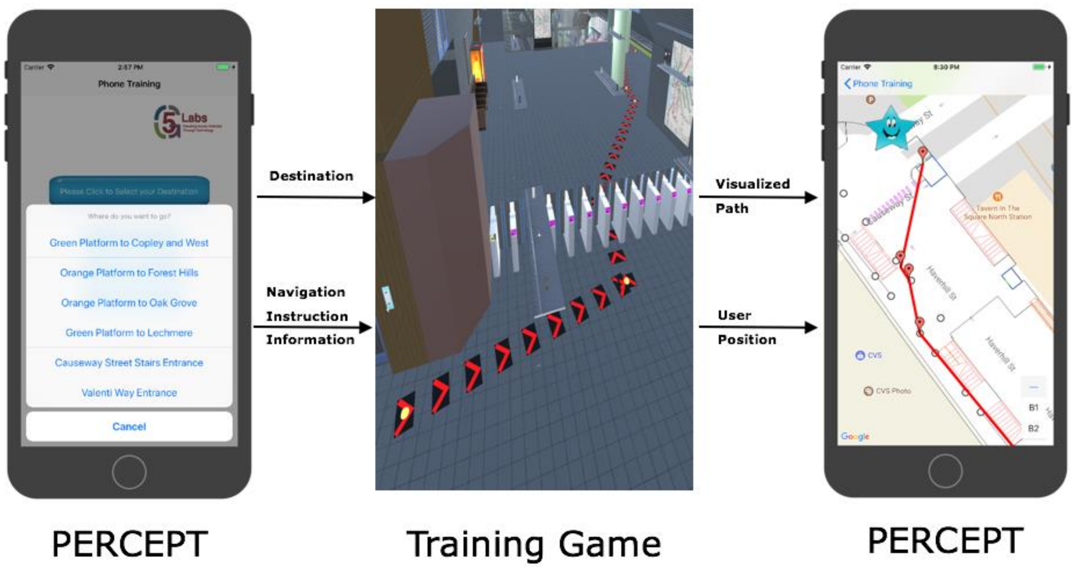

Figure 4.25: Interaction between PERCEPT and training game.

We created a PERCEPT shell application for testing. In the shell application, the user can choose their destination and user preference, then the related path information and navigation instructions will be loaded from the PERCEPT web database. Data processing will be done on the shell application side, including setting the help nodes and translating the coordinates. These processed data will be sent into pre-travel training game synchronously. At this time, the game will not get any information directly from PERCEPT web database but only from the shell application. After the game received the complete data package, it will generate a visible path depends on the received data. The user could see this visualized path as well as on the shell application’s 2D map. Then the user can control the avatar to explore the environment in the game using the touchpad. The blue star in PERCEPT application indicates the user’s position when player moving inside the game its position information will send back to PERCEPT synchronously.

CHAPTER 5

REAL-TIME GUIDANCE

After the user feels confident with the virtual environment, they can go to the physical space (North Station). In order to keep assisting the user even they are in the physical environment, we deployed the game on the smartphone to provide real-time guidance. In the pre-travel training game, the avatar is controlled by the user using two 2 joysticks to move. Consider when user in the physical environment using joysticks is very unsafe and inconvenience, we provide another way to control the avatar's move. Our game will calculate user's position in real-time and mapping it into the corresponding position inside the game continuously, in that way the avatar can move based on user's natural walking in physical environment. Three phases of work have been done to achieve this. Firstly, we leveraged the work which finished in Step 1 pre-travel training module, using the training game as a virtual environment in real-time guidance module and convert the laptop version game to mobile version game which can launch on the smartphone. Secondly, we integrated real-time indoor localization to track the user by providing their position information continuously. Finally, we created a user interface for the real-time guidance module.

5.1 Virtual Environment

We use the training game which has been developed in the pre-travel training module as the environment in real-time guidance module. Here are the steps to convert a game from laptop version to iOS version:

Step1: In Building Settings window, click the "Switch Platform" button to switch to "iOS" from the original "Standalone" platform.

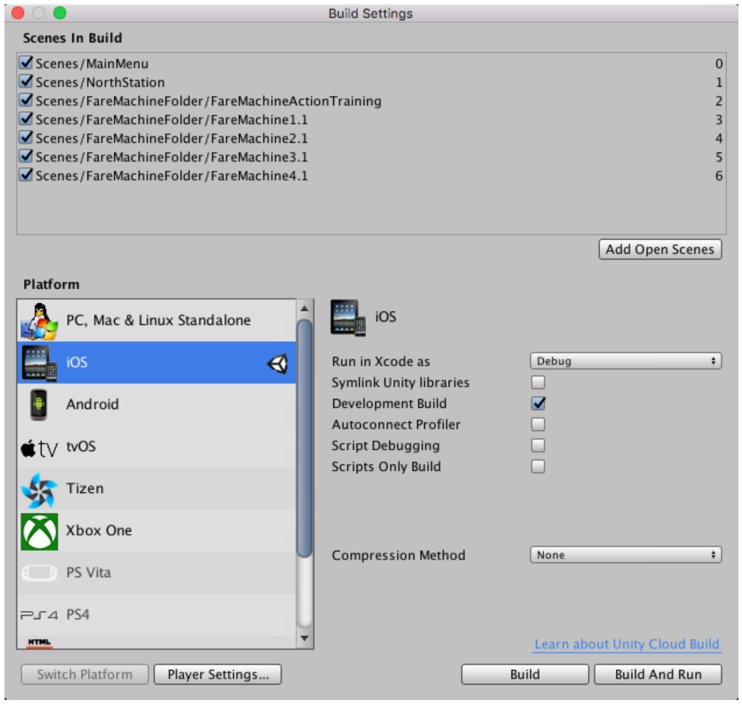

Figure 5.1: Build setting window.

Step2: Set the player settings specific to iOS version.

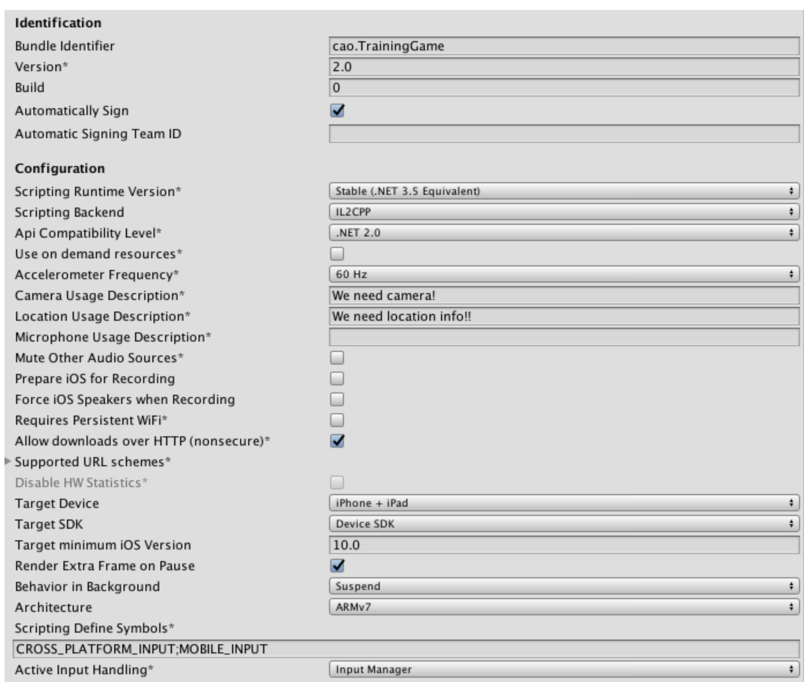

Figure 5.2: Player setting window.

Step3: Setup a dual touch control on the mobile phone. Two touch buttons inside the red rectangular is the dual touch control button. The left one is using for moving, and the right one is using for observing around.

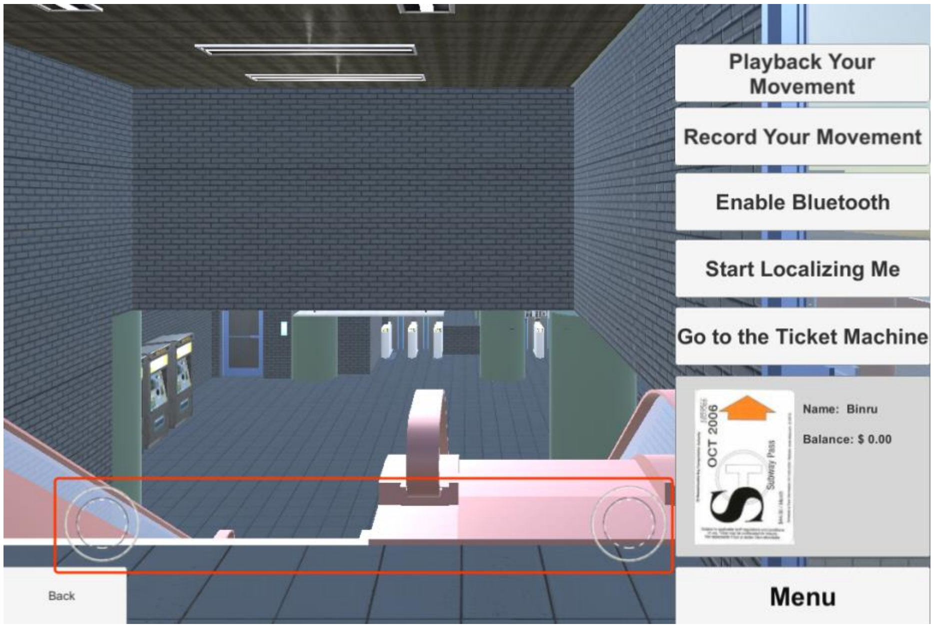

Figure 5.3: Dual touch control for touch input.

Step4: Build the game into Xcode and launch it on iOS device.

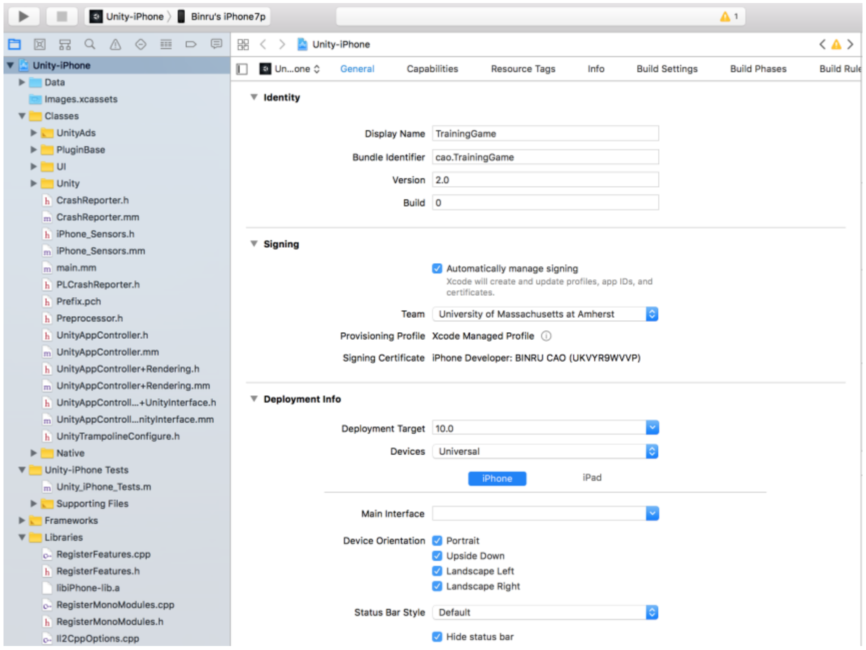

Figure 5.4: Project in Xcode

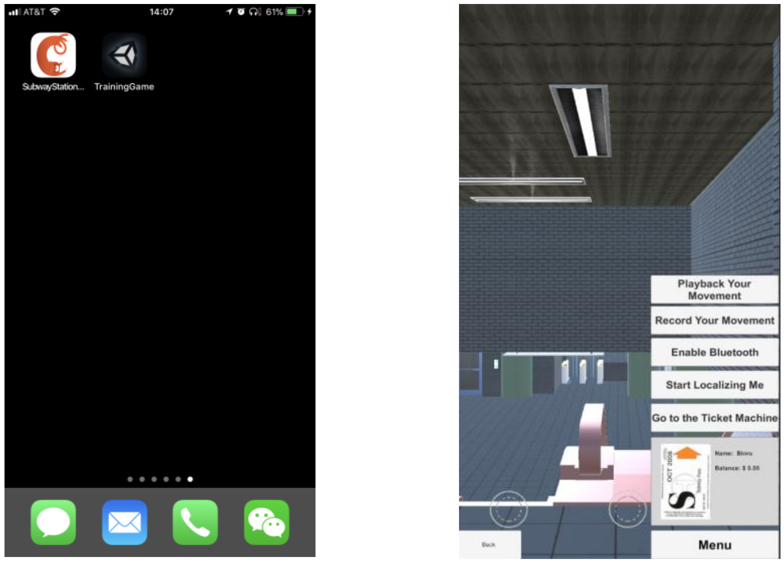

Figure 5.5: Screen shot for the game launching on smartphone.

5.2 Real-time Indoor Localization and Beacon Deployment

After we launched the mobile version game on a mobile device which supports BLE technology, we can add the BLE-based indoor localization function into the guidance assistance module to provide a real-time indoor position of the user and track their footprint when they were moving.

The advent of Bluetooth Low Energy (BLE) technology gave opportunities for immense improvement in indoor positioning. It promotes a low cost and easily deployed indoor localization solution. It only needs a Bluetooth enabled device like a mobile phone and some BLE-powered devices like Beacons (BLE tags).

5.2.1 BLE-based Indoor Localization

The concept of BLE-based localization is used different RSSI signals send from multiple beacons at different places to calculate an intersecting point. Firstly, we need to convert the RSSI signal to a calculatable distance value. RSSI signals are electromagnetic waves that propagate through a transmission media; the significant feather of transmission is the signal strength decreases as the distance increases. Based on the signal strength, the distance can be calculated by the Path-loss Propagation model.

$$RSSI_d = RSSI_{d0} + 10 * \gamma * \log_{10} \frac{d}{d0} \tag{5.1}$$

$RSSI_d$: The received signal strength at distance d

$RSSI_{d0}$: The received signal strength at distance $d0$, usually $d0$ = 1 meter. In our system the value of $RSSI_{d0}$ is -63.5379 dB

γ : The path loss index depends on the propagation environment, in our system the value is -2.086

From equation (5.2), we can calculate the corresponding distance by:

$$d = 10^{\left(\frac{RSSI_d + 63.5379}{-20.86}\right)} \tag{5.2}$$

Secondly, we apply the classical indoor positioning method called “Trilateration” in our system. For the indoor location system, 2D trilateration is used to find a tagged object that is located on a surface, which will be on an XY plane [14]. Since we only use the latitude and longitude information of beacons our location system is the XY plane-based system.

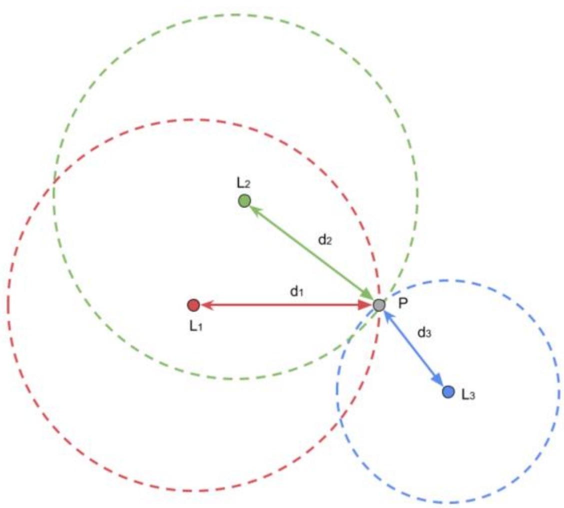

Figure 5.6: P: unknown point. Li: reference points.

As Figure 5.6 shown, each reference point has its own position, expressed with latitude and longitudes (x1, y1), (x2, y2), (x3, y3). The target (unknown) point will simultaneously satisfies the equation of these three circles:

$$(x-x1)^2 + (y-y1)^2 = d1^2 \tag{5.3}$$

$$(x-x2)^2 + (y-y2)^2 = d2^2 \tag{5.4}$$

$$(x-x3)^2 + (y-y3)^2 = d3^2 \tag{5.5}$$

Once we use these equations to get the distance (di) between the target point and reference point, we can easily calculate the latitude and longitude of the target point by solving these equations. In order to apply this positioning method in our system, we must make sure that for every single point in the navigation area can be covered at least three beacons. Otherwise, we cannot localize a single intersection point with only two circles. Therefore, I designed a beacon deployment to satisfy our positioning system.

5.2.2 Beacon Deployment

The goal is to find a solution to deploy BLE beacons in the service area to provide the required RSSI values. Consider the cost of beacons, the optimum deployment should use a minimum number of beacons and provide reasonable accurate localization results.

Because of the site restriction, I use Campus Center of the University of Massachusetts Amherst as the test environment instead of Boston North Station. The service area shows below as the pink square and to simplify the calculation I assumed the target nodes are a set of nodes with a grid spacing of 1ft that uniformly distributed in the service area and the signal from every beacon has the same propagation.

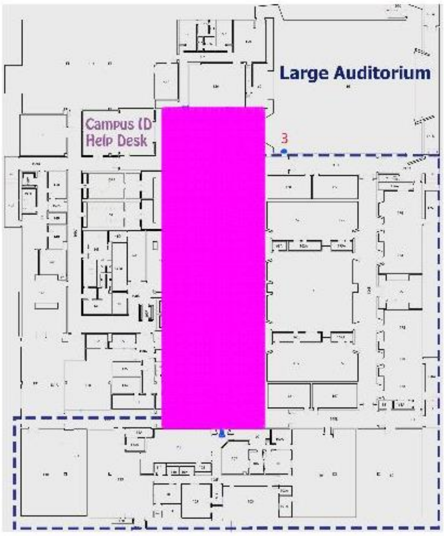

Figure 5.7: The size of service area is 54.4ft × 172.4ft.

The structure of the deployment is shown below using input-output framework.

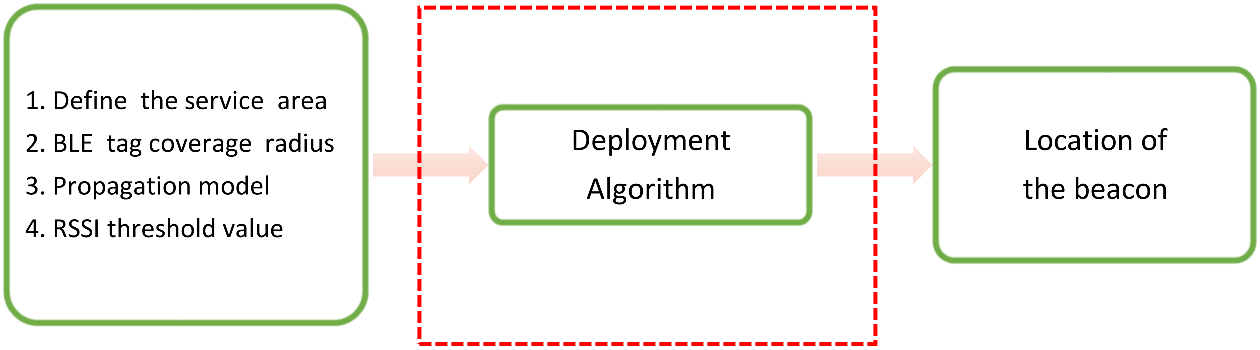

Figure 5.8: The flow chart for beacon deployment.

- Deployment Algorithm

I use the *Minimize-Number of Reference Node Model (MNR)* [15] and the *Binary Integer Linear Programming (BILP)* [16] to implement the deployment.

The MNR model aims to deploy the minimize the number of RNs (Reference nodes, in our system are beacons) in the environment such that any target node within the service area can communicate with at least n RNs (in our case, the n will be 3). This can be written as an objective function as:

$$N_{MNR} = \ Minimize \sum_{\forall j \,\epsilon\, R} C_j \tag{5.6}$$

N_{MNR} is a constant parameter means the optimal number of beacons obtained from the deployment algorithm. R is a set of candidate sites to install beacons. C_j is a binary {0, 1} variable that equals 1 if the beacon is installed at site j, $j \,\epsilon\, R$; 0 otherwise. To select beacon installed locations from a set of predetermined candidate locations can be incorporated into the model via several constraints inequation:

$$\sum_{\forall j \,\epsilon\, R} C_j \geq N_R \tag{5.7}$$

$$S_{ij}\left(P_{ij} - P_T\right) > 0 \tag{5.8}$$

$$S_{ij} \leq \ C_j \tag{5.9}$$

$$\textstyle\sum_{\forall j \,\epsilon R} S_{ij} \geq \ N_R \tag{5.10}$$

N_R is the minimum number of beacons that cover each target point, in our case this number is at least 3. Constraint (5.7) states that the minimum installed beacons in the environment is at least equal to N_R . P_T is the received signal strength threshold. We set it in a range of -80dB to -90dB which makes sure the good communication quality in most cases. P_{ij} is the RSSI signal strength that a target point i received from beacon j, $i \,\epsilon\, T$, T is a set of target points. S_{ij} is a binary

{0, 1} variable that equals 1 if the target point i is assigned to beacon j, 0 otherwise. Constraint (5.8) states that target node i is in coverage of beacon j if the signal strength received by node i from beacon j is larger than the threshold. Constraint (5.9) make sure that target node i can receive the signal from beacon j only if beacon j is installed. Constraint (5.10) enforces that target node i must receive the signals from at least N_R beacons.

- Implementation

I use Matlab to implement this deployment algorithm and build a User Interface to change the parameters in a defined service area. The program has a GUI screen that allows the user to set the correlative parameters to have the deployment result. The changeable parameters are 1. Beacon coverage radius. 2. RSSI signal threshold. 3. The number of beacons that cover each target node. I had test 4 cases based on different assumptions and requirements.

Case 1

Assumptions/Inputs:

- The service area: Length is 181.02 ft. Width is 57.12 ft.
- The path-loss propagation model is set as:

$$RSSI = distance - 63.5379 + 10 * (-2.086) * log10(distance/3)$$

- Each target node covered by at least 3 beacons
- Beacon coverage radius is 15ft
- RSSI threshold value is -85dB

Result:

- The final number of the beacons is 27
- The locations show below (The green nodes are candidates of beacons; the blue nodes are final beacons):

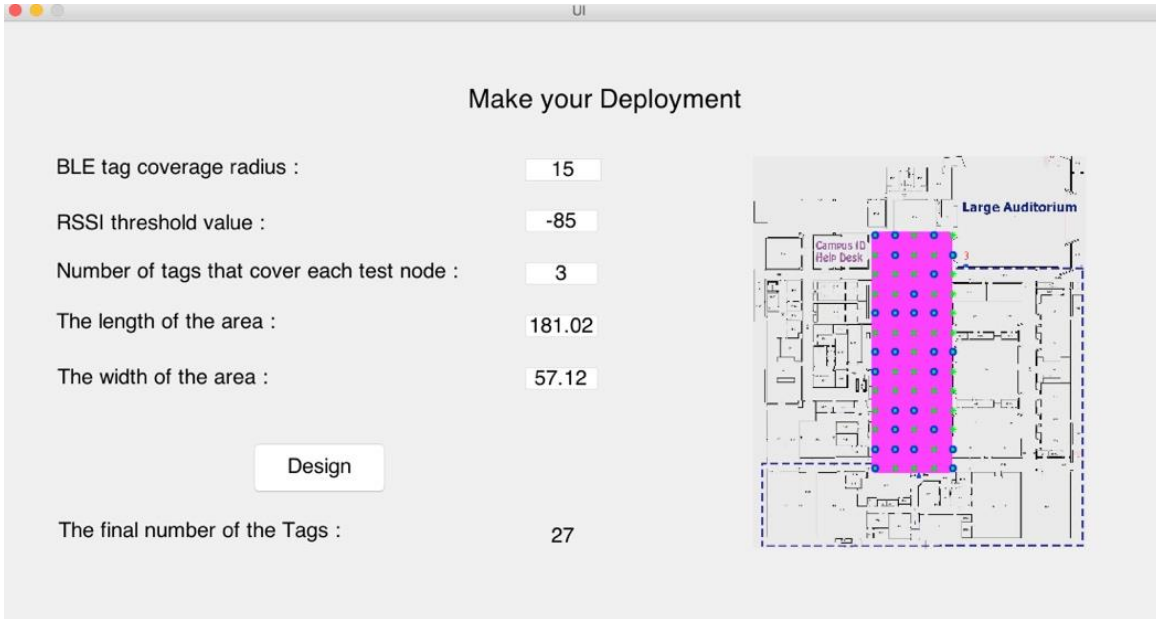

Figure 5.9: Results in Case1.

Case 2

Assumptions/Inputs:

- The service area: Length is 181.02 ft. Width is 57.12 ft.
- The path-loss propagation model is set as:

$$RSSI = distance - 63.5379 + 10 * (-2.086) * log10(distance/3)$$

- Each target node covered by at least 4 beacons
- Beacon coverage radius is 20ft
- RSSI threshold value is -85dB

Result:

- The final number of the beacons is 35
- The locations show below (The green nodes are candidates of beacons; the blue nodes are final beacons):

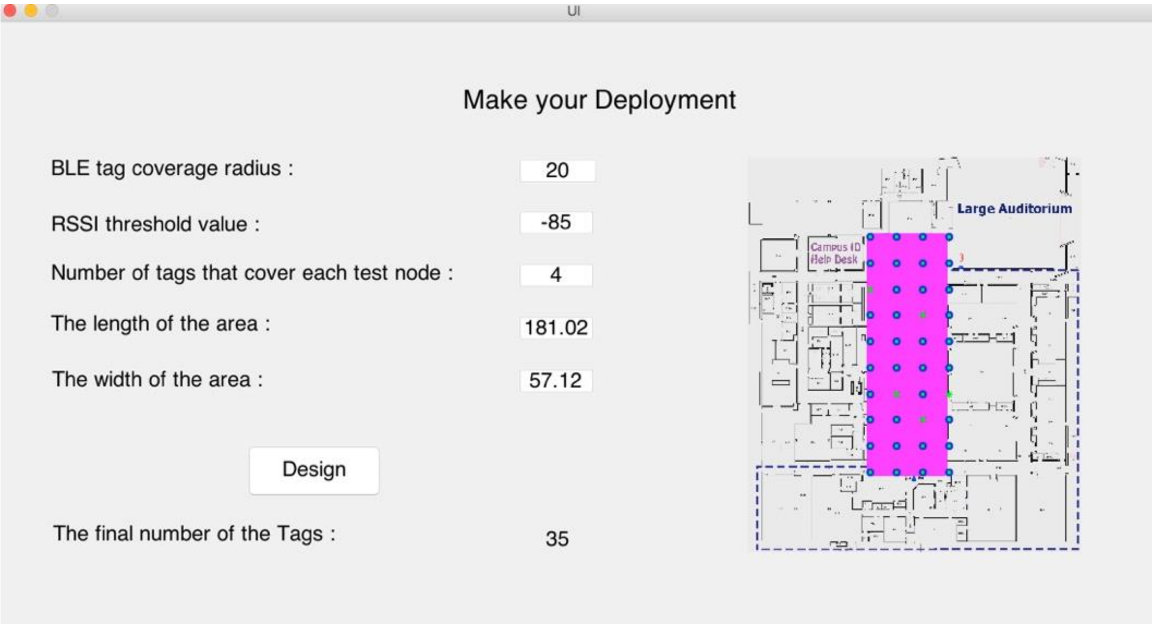

Figure 5.10: Results in Case 2.

Case 3

Assumptions/Inputs:

- The service area: Length is 181.02 ft. Width is 57.12 ft.
- The path-loss propagation model is set as:

$$RSSI = distance - 63.5379 + 10 * (-2.086) * log10(distance/3)$$

- Each target node covered by at least 4 beacons
- Beacon coverage radius is 20ft
- RSSI threshold value is -90dB

Result:

- The final number of the beacons is 15
- The locations show below (The green nodes are candidates of beacons; the blue nodes are final beacons):

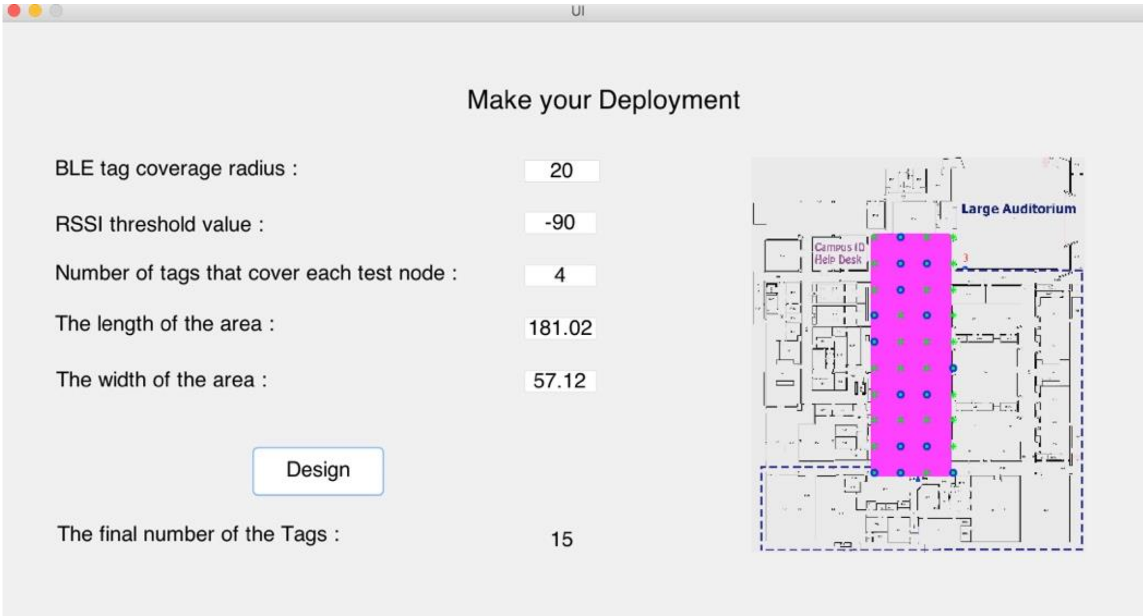

Figure 5.11: Results in Case3.

Case 4

Assumptions/Inputs:

- The service area: Length is 181.02 ft. Width is 57.12 ft.
- The path-loss propagation model is set as:

$$RSSI = distance - 63.5379 + 10 * (-2.086) * log10(distance/3)$$

- Each target node covered by at least 3 tags
- Beacon coverage radius is 15ft
- RSSI threshold value is -90dB

Result:

- The final number of the beacons is 12

- The locations show below (The green nodes are candidates of beacons; the blue nodes are final beacons):

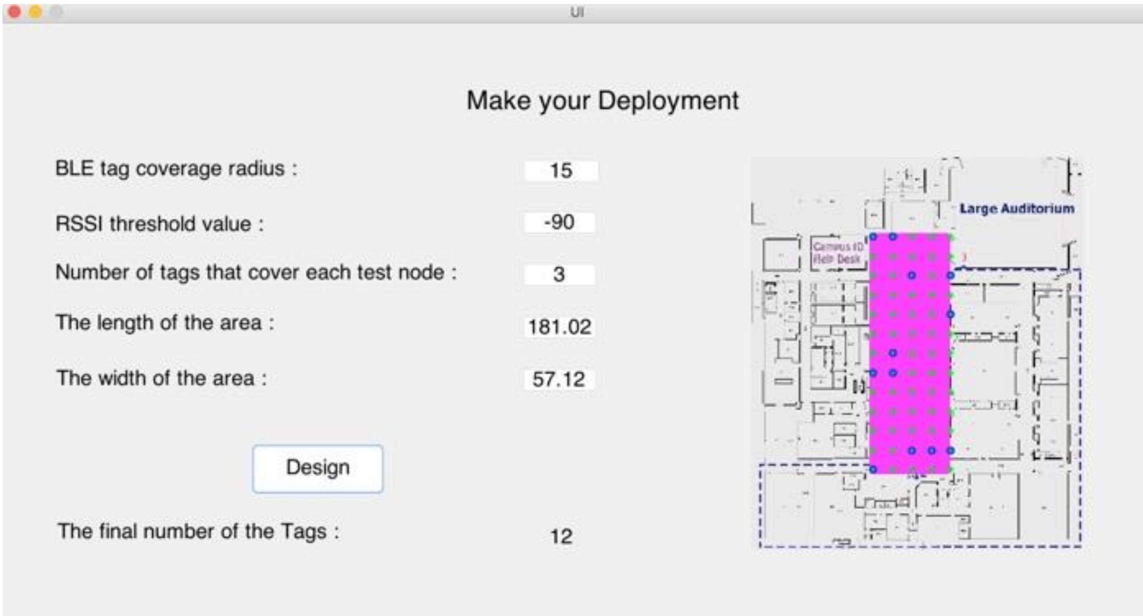

Figure 5.12: Results in Case 4.

- Test in Campus Center

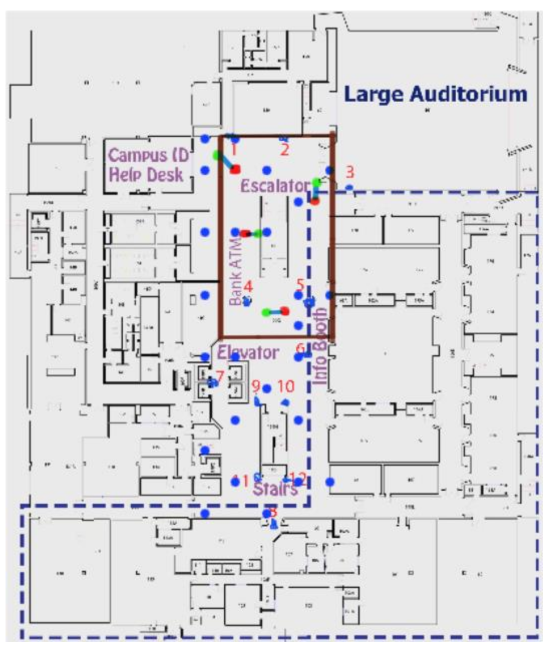

Figure 5.13. Results in Campus Center.

I tested the beacon deployment solution at Campus Center. I use the deployment as case 2, due to the restriction of the building's arrangement the actual deployment is little different from the simulated deployment. The brown area is the test space. The red nodes show the real location (ground truth) and the green nodes indicate the calculated result showing on the application.

Consider we already have the beacons deployed in North Station for PERCEPT use and change the deployment in Boston North Station is a large and challenging project, I decided to use the existing deployment when the test in North Station since it has the same design idea with my deployment.

5.3 Real-time navigation guidance in North Station

After we find the appropriate localization algorithm and beacon deployment solution, we can apply them into our navigation system to provide a real-time guidance assistant when the user in the physic environment and need get some navigation help.

We use the iBeacon plugin in Unity that supports the system to detect and receive beacons. We create a new iBeaconReceiver game object that allows the system to receive RSSI signals. Every single beacon has a set of attributes include Region name (which is a unique identifier), Type (a definition of the beacon which is associated, our beacons are the type of iBeacon), Proximity UUID (the unique ID of the beacons it targets), Major id (the major value that we use to identify one or more beacons) and Minor id (the minor value that we use to identify a specific beacon). In our service environment, all the beacons share the same Region name (Kontakt), Type (iBeacon), UUID (401EE3FD-6F9F-4500-8F47-A99D25C66412) and Major id (4), the only difference is the Minor id. We identify a specific beacon using Major-Minor value. In the iBeaconReceiver, the Major and Minor ids are set to 0 to make sure our system can detect all beacons that have the same UUID.

We also attached a Bluetooth State component on this iBeaconReceiver game object that works with the Bluetooth on the device. On iOS, the application needs the Location authorization and Bluetooth Peripheral Usage authorization from the user to have permission to access the beacons at any time. We set it in Xcode customer setting when compile and build the application. The iBeaconReceiver object will first detect the status of BLE on the device. If the Bluetooth is turned on it can start to scan the detectable beacons and get the Major-Minor values and RSSI signals.

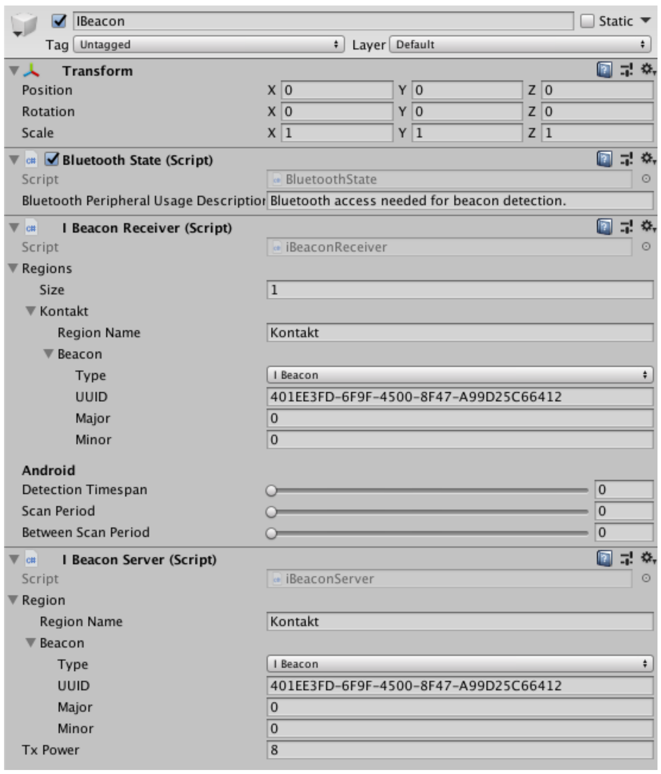

Figure 5.14: iBeacon component in Unity.

Figure 5.15 shows the work flow of our indoor localization function. The iBeaconReceiver object receives and updates the data every time a beacon is detected. We can process these data at the same time to provide a localization result and translate into corresponding Unity environment synchronously.

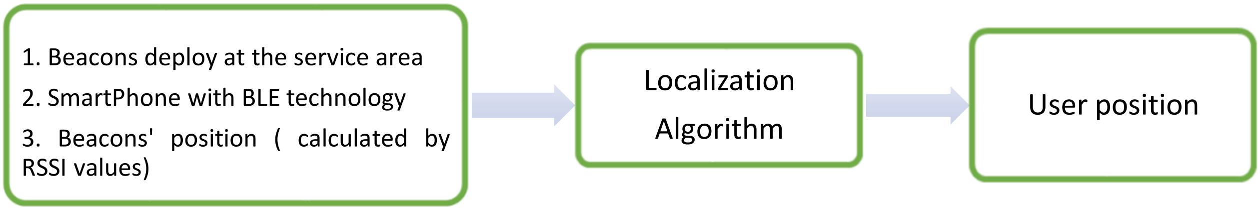

Figure 5.15: Work flow of indoor localization

When the user moves in the physical environment, the avatar in the game environment will follow the same path and move in the same direction automatically. The first-person point of view in the game will accurately indicate the scene of the real world. With the navigation instruction inside the game environment, the user can easily find where they are, and which direction should they forward to find the correct path to their destination.

5.4 Testing Result

We take the test in Boston North Station. Figure 5.16 is the testing route we followed in Boston North Station. We started from fare gate (paid side) at Valenti entrance lobby and ended at the elevator located in the middle of green line towards Lechmere. The totally length of the testing route is about 150ft. Figure 5.17 is the screenshot of the result video. The Left side is our game scene, the right side is the ground truth video. We follow the navigation instruction showing in the game and try to move follow the sign. The localization function will localize the users at the same time and control the avatar's movement based on the localized results in the game environment. We also take a video about the path we walked at the same time as the standard ground truth to evaluate our results.

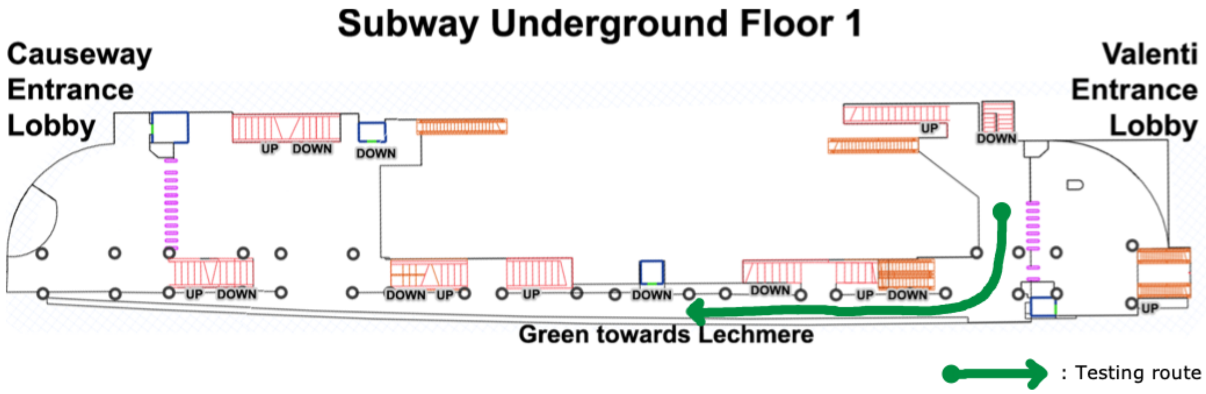

Figure 5.16: Testing route in Boston North Station.

The result shows that our localization system works well in the North Station. Compared with other navigation systems with only have 2D map about the environment, our system can provide a visualized view about the surroundings. The user can get navigation assistant by following the highlight navigation instruction signs. The game view can synchronously indicate the real-time environment in the physic environment that the user located in. In order to provide a smooth and stable view for the users, we split the environment into several small regions and set a distance threshold. The average deviation of our real-time navigation system is about 15ft-20ft.

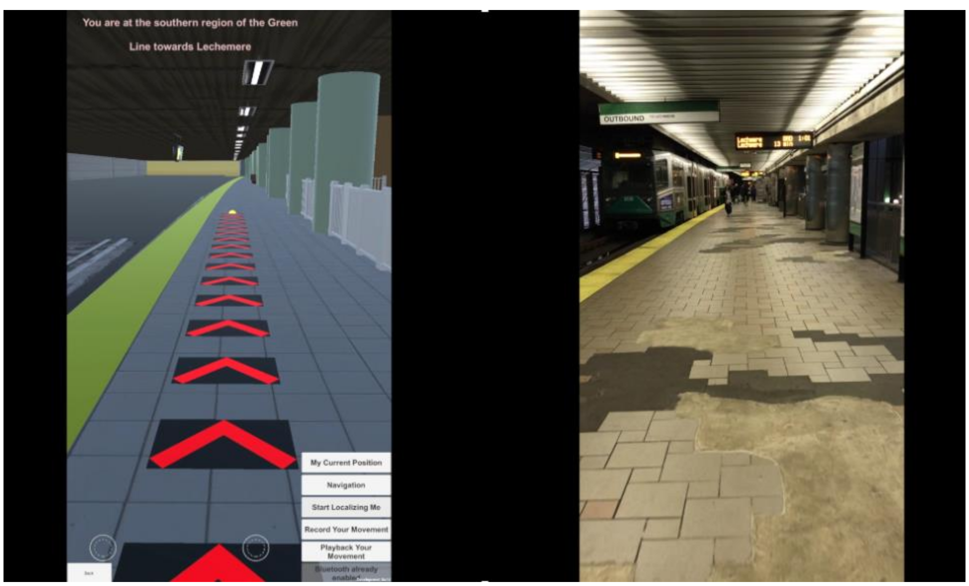

Figure 5.17: Test result in Boston North Station.

CHAPTER 6

CONCLUSION AND FUTURE WORK

In this paper, we introduce a two-step navigation assistant system includes an offline pre-travel training module and a real-time navigation guidance module. The system can help the users get familiar with surroundings and practice how to follow the navigation instruction to arrive a presupposed destination before they go to the strange environment, and it also provides a simple and intuitively guidance when the user in the real-time environment.

The virtual environment in our system is now only built for the North Station. In the future, we can apply it universally to other stations. Meanwhile, the current localization method using in our system is BLE-based algorithm. In some terrible situations, this method might not provide a highly accurate positioning result, to solve this problem our current system uses regions to provide the smooth view. In the future, we can consider introducing more positioning methods use different technologies like RFID-based localization system, Magnetic position system and Wi-Fi-based position system. By combining them together, the system can switch the different positioning functions based on the situation in environment to cover each’s shortages. The system can sign a weight factor for each function then calculate the final localization positions to provide a highly accurate localization result.

BIBLIOGRAPHY

[1] H. A. Karimi, Indoor Wayfinding and Navigation, CRC Press, 2015.

[2] D. G. a. P. K. Wan He, "An Aging World: 2015 International Population Reports," U.S. Census Bureau, 2016.

[3] W. H. Organization, "World report on disability," [Online]. Available: http://www.who.int/disabilities/world_report/2011/report/en/.

[4] J. B. J. F. Ramon Hervas, "An Assistive Navigation System Based on Augmented Reality and Context Awareness for People With Mild Cognitive Impairments," pp. 368-374, Jan 2014.

[5] S.-K. T. T.-Y. W. Yao-Jen Chang, "A Context Aware Handheld Wayfinding System for Individuals with Cognitive Impairments," in *10th international ACM SIGACCESS conference on Computers and accessibility*, 2008.

[6] Y.-J. Chang, C.-N. Chen, L.-D. Chou and T.-Y. Wang, "A Novel Indoor Wayfinding System Based on Passive RFID for Individuals with Cognitive Impairments," in *2008 Second International Conference on Pervasive Computing Technologies for Healthcare*, 2008.

[7] S.-K. T. T.-Y. W. Yao-Jen Chang, "A Context Aware Handheld Wayfinding System for Individuals with Cognitive Impairments," in *10th international ACM SIGACCESS conference on Computers and accessibility*, 2008.

[8] X. A. G. M. J. C. T. A. P. Javier GÓMEZ, "AmICog – mobile technologies to assist people with cognitive disabilities in the work place," *ADVANCES IN DISTRIBUTED COMPUTING AND ARTIFICIAL INTELLIGENCE JOURNAL*, vol. 2, 2013.

[9] P. Z. A. Mancini, "Point to Point Navigation for People with Mobility Impairments," in *2014 IEEE/ASME 10th International Conference on x Mechatronic and Embedded Systems and Applications (MESA)*, 2014.

[10] L. C. d. Oliveira, A. O. Andrade, E. C. d. Oliveira, A. Soares, A. Cardoso and E. Lamounier, "Indoor Navigation with Mobile Augmented Reality and beacon technology for Wheelchair users," in *2017 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*, 2017.

[11] R. Fleiner, G. Simon-Nagy and B. Szász, "Accessible Indoor Navigation based on Linked Data in Hospitals," in *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2016.

[12] Wikipedia, "Unity (game engine)," 16 Oct 2017. [Online]. Available: https://en.wikipedia.org/wiki/Unity_(game_engine).

[13] Unity, "Unity Manual," Unity, [Online]. Available: https://docs.unity3d.com/Manual/ScriptingSection.html.

[14] A. M. K. T. B. Z. Robert Jarvis, Indoor Positioning System, 11/29/2011.

[15] K. K. C. P. K. K. Supattra Aomumpai, "“Optimal Placement of Reference Nodes for Wireless Indoor Positioning Systems”," in *Electrical Engineering /Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2014 11th International Conference*, 2014.05.

[16] D. T. a. A. J. A. Bari, "“Optimal relay node placement in hierarchical sensor networks with mobile data collector,”," in *Computer Communications and Networks Proceedings*, 2009, pp. 1-6..

[17] A. T. J. K. A. V. Ruquia Mirzll, "An Indoor Navigation Approach to Aid the Physically Disabled People," in *International Conference on Computing, Electronics and Electrical Technologies*, Kumaracoil, 2012.

[18] A. B. A. B. C. V. D. J. P. A. a. S. G. Akhila Kishore, "CENSE: A Cognitive Navigation System for People with Special Needs," in *IEEE Third International Conference on Big Data Computing Service and Applications*, 2017.

[19] D. D. D. V. N. F. D. P. a. R. V. D. W. S. Verstockt, "Assistive smartphone for people with special needs : The Personal Social Assistant," in *2009 2nd Conference on Human System Interactions*, 2009.

[20] National Institute on Aging (NIA), "World's older population grows dramatically," 28 March 2016. [Online]. Available: https://www.nih.gov/news-events/news-releases/worlds-older-population-grows-dramatically.

[21] J. F. Ian Spence, "Video Games and Spatial Cognition," *American Psychological Association,* 2010.

[22] Wikipedia, "Electric_beacon," 19 September 2017. [Online]. Available: https://en.wikipedia.org/wiki/Electric_beacon.

[23] K. K. C. P. K. K. Supattra Aomumpai, "Optimal Placement of Reference Nodes for Wireless Indoor Positioning Systems," in *Electrical Engineering /Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2014 11th International Conference*, 2014.

[24] G. M. J. C. T. A. P. Javier Gomez, "An Adapted Wayfinding System for Pedestrians with Cognitive Disabilities," *Mobile Information Systems,* vol. 2015, 2015.

[25] L. Li, Q. Xu, V. Chandrasekhar, J.-H. Lim, C. Tan and M. A. Mukawa, "A Wearable Virtual Usher for Vision-Based Cognitive Indoor Navigation," *IEEE Transactions on Cybernetics,* vol. 47, no. 4, pp. 841-854, 2017.

[26] R. Hervás, J. Bravo and J. Fontecha, "An Assistive Navigation System Based on Augmented Reality and Context Awareness for People with Mild Cognitive Impairments," *IEEE Journal of Biomedical and Health Informatics,* vol. 18, no. 1, pp. 368-374, 2014.

[27] A. Covaci, D. Kramer, J. C. Augusto, S. Rus and A. Braun, "Assessing Real World Imagery in Virtual Environments for People with Cognitive Disabilities," in *2015 International Conference on Intelligent Environments*, 2015.

[28] W. H. S. C. B. Benjamin Poppinga, "NavMem Explorer- An Orientation Aid for People with Mild Cognitive Impairments," in *Proceedings of Pervasive Health*, 2014.

[29] L. Ding, "Navigation Instruction Validation Tool and Indoor Wayfinding Training Application for People with Disabilities," 2017.